5-10-2021

Assembly and annotation of the genome of an invasive bush honeysuckle, Amur honeysuckle (Lonicera maackii)

Erin R. Kesel
erk2575@rit.edu

RIT

Assembly and annotation of the genome of an invasive bush honeysuckle, Amur honeysuckle (*Lonicera maackii*)

By

Erin R. Kesel

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Bioinformatics

Department of Bioinformatics

College of Science

Thomas H. Gosnell School of Life Sciences

Rochester Institute of Technology

Rochester, NY

May 10, 2021

Rochester Institute of Technology
Thomas H. Gosnell School of Life Sciences
Bioinformatics Program

To: Head, Thomas H. Gosnell School of Life Sciences

The undersigned state that Erin Kesel, a candidate for the Master of Science degree in Bioinformatics, has submitted her thesis and has satisfactorily defended it.

This completes the requirements for the Master of Science degree in Bioinformatics at Rochester Institute of Technology.

Thesis committee members:

Name	**Date**
Michael V. Osier, Ph.D. Thesis Advisor	
Eli J. Borrego, Ph.D.	
André O. Hudson, Ph.D.	
Susan S. Pagano, Ph.D.	

Feng Cui, Ph.D.
Director of Bioinformatics MS Program

475-4115 (voice)
fxcsbi@rit.edu

ABSTRACT

Invasive species are a global problem that cause significant environmental and economic damage. It has been estimated that the cost of invasive species in the United States is at least 120 billion dollars annually. *Lonicera maackii*, commonly known as Amur honeysuckle, is an invasive shrub found in New York State. In invaded regions, *L. maackii* has caused decreased species richness of native plant and animal species. It has also negatively affected the native migratory bird populations that eat the plant's berries. Currently, there is no available genomic sequence for any *Lonicera* species. With next generation sequencing, new information can be unveiled that can inform control strategies and provide a better understanding of *L. maackii* as an invasive species. In this study, a genome sequence was assembled for an individual of *Lonicera maackii* found in Western New York. The assembled genome was annotated using two different methods. Genes found through annotation provide direction for future work on optimum control strategies for the *L. maackii* invasion.

TABLE OF CONTENTS

TABLE OF FIGURES AND TABLES

INTRODUCTION

Bush honeysuckles were originally brought to North America from Europe and Asia in the 1800s as ornamental shrubs (Smith and Smith, 2010). Over the last century, non-native bush honeysuckles have escaped cultivation and are causing multifaceted consequences for local ecosystems. Invasive honeysuckles displace native shrubs, crowd and shade out tree saplings, outcompete native species for resources, and reduce species richness of local plants and animals (Smith and Smith 2010; Watling et al., 2011; Hudon et al., 2017). Mechanical, chemical, and combination control measures have been implemented, but the ability of bush honeysuckles to thrive in North America makes effective control a challenge. While a lot of research has been done to understand bush honeysuckles and their role in local ecosystems, little is known about bush honeysuckles on the genomic level. The use of next generation sequencing technologies could provide insight into effective control strategies and how to mitigate some of the ecological consequences of these invasive plants.

Invasive species

An invasive species is characterized as a non-native species that can cause harm to the economy, the environment, and or human health (Beck et al., 2008). A species is considered to have invaded an area when its spread to non-native areas is uncontrolled or unintentional. One characteristic of an invasive species is its increased success in a non-native habitat, a phenomenon known as "invasion success" (Prior et al., 2015). The enemy release hypothesis provides one explanation for invasion success. According to the enemy release or natural enemy hypothesis, the success and ability of a non-native species to invade an area depends on the absence or reduction of natural limiting factors in the new habitat (Maron and Vila, 2001). Without natural predators,

the invasive species outcompetes native species for available resources, allowing the invasive to flourish while natives struggle to survive.

Invasive species have negative ecological and economic consequences. One of the most pressing issues caused by the introduction of an invasive species is loss of local biodiversity. Success of invasive species is often coupled with the downfall of many native species. For example, in Lake Victoria, Africa, the invasive fish Nile Perch (*Lates niloticus*) preyed upon and outcompeted local fish species leading to the extinction of over 200 native species (Lowe et al., 2000). In addition, controlling invasive species requires a great deal of resources. Time and money must be diverted away from other important efforts to control invasives and to fix the problems that the invasives cause. In a study of yellow starthistle, an invasive weed, it was estimated that the direct and indirect yearly costs because the yellow starthistle was 12.7 million dollars (Julia et al., 2007).

Invasive species have also been shown to negatively impact human health and wellbeing. In a study to assess the impact of invasive honeysuckle removal, untreated plots were found to harbor more mosquitoes that were vectors for West Nile virus, suggesting that the presence of invasives can promote vector-borne diseases (Gardner et al., 2017). Because of their widespread negative impacts, understanding and controlling invasive species is a current topic of interest.

Lonicera maackii

Lonicera maackii (Rupr.) Herder (Caprifoliaceae), commonly known as Amur honeysuckle, bush honeysuckle, tree honeysuckle, or Maack's honeysuckle, is an invasive deciduous shrub native to parts of Asia (Luken and Thieret, 1996). *L. maackii* was originally brought from eastern Asia to North America as an ornamental shrub in the late 1800s. Its flowers and fruit were very popular among gardeners. *L. maackii* is an upright, multi-stemmed shrub that

can grow up to six meters tall, spread nine meters across, and has stems with diameters up to fifteen centimeters (Luken and Thieret, 1995). The stems emerge at the ground from a central woody burl (Deering and Vankat, 1999). The bark on the stems is gray-brown and has longitudinal fissures. The shrub has dark green, acuminate leaves that have an average length of seven centimeters. In early spring, the plant produces white to pink paired axillary flowers that fade to dark yellow with age. In the fall, 3.8-8.5 millimeter glossy red or orange berries ripen and may remain until late December (Luken and Thieret, 1995). *L. maackii* has a diploid chromosome count of eighteen (Ammal and Saunders, 1952).

Four species of invasive bush honeysuckle that are found in the United States are *L. tatarica* (Tartarian honeysuckle), *L. morrowii* (morrow's honeysuckle), *Lonicera x bella* (Bella or Showy honeysuckle), and *L. maackii* (Czarapata, 2005). These species are difficult to distinguish because they have many similar characteristics, and species crosses can also exist (Czarapata, 2005). Physical characteristics of the four bush honeysuckle species are summarized in Table 1. All four of these *Lonicera* species are multi-stemmed, upright shrubs, have shallow root systems, produce flowers and red fruit (Czarapata, 2005). One distinguishing feature of *L. maackii* is its leaves; they are larger and have a longer point than other bush honeysuckle species (Czarapata, 2005). Tatarian honeysuckle has bluish-green leaves, but the leaves of the other three species are dark green. Morrow's honeysuckle leaves are much narrower than those of Amur honeysuckle (Czarapata, 2005). The flowers of Tatarian and Bella's honeysuckle are much pinker than those of Morrow's and Amur honeysuckle (Czarapata, 2005). *L. maackii*, which reaches a maximum height of eighteen feet, can grow larger than both Morrow's and Tatarian honeysuckles, which grow to a maximum height of six and nine feet respectively (Czarapata, 2005).

Table 1. Distinguishing characteristics of bush honeysuckle species.[1]

Species	*L. maackii*	*L. morrowii*	*L. tatarica*	*L. x bella*
Height	Up to 18 feet	Up to 6 feet	Up to 9 feet	Up to 18 feet
Flower color	White to pink, yellow with age	White, yellow with age	Pink or purple-red	Pink, yellow with age
Flower peduncle	0.2 inches or shorter	0.2 to 0.6 inches long	0.6 to 1 inch long	0.2 to 0.6 inches long
Leaves	Elliptical, 1.5 to 3.5 inches long, narrow, long point	Elliptical to oblong, pointed tips, 1 to 2.5 inches	Oval to oblong, hairless, 1 to 2.5 inches, blue-green	Elliptical, oblong, or oval, 1 o 2.5 inches long

[1]Information from Czarapata, 2005.

L. maackii reproduces via seed dispersal. Native animals, including birds and deer, consume the bush's berries and disperse the seeds (Bartuszevige & Gorchov, 2006, Castellano & Gorchov, 2013). *L. maackii* does not reproduce through vegetative reproduction (Luken and Goessling, 1995). Individuals reach reproductive maturation three to eight years after sprouting. However, seed production has been shown to be dependent on plant height, not age (Deering and Vankat, 1999). Energy of young *L. maackii* plants is devoted to crown expansion. During the pre-reproductive years, increases in height and the number of stems is observed with minimal increase in basal area (Deering and Vankat, 1999). Once individuals reach reproductive age, the pattern of growth changes. The number of basal stems remains relatively constant at four, older stems are maintained, and radial growth leads to an increase in basal area. Stem basal area increases exponentially with rapid growth beginning round five years whereas height increases uniformly with age.

L. maackii is native to China, Korea, and Japan (Luken and Thiret 2003). In its native region, *L. maackii* primarily occupies floodplains and open woodlands (Luken and Thiret 2003).

However, it has been found to invade almost any habitat in North America, including abandoned fields, pastures, the edge of woodlands, floodplains, the edge of highways and railways, vacant lots, and gardens (NYISI, 2019). While *L. maackii* grows best in full sun, it is semi shade-tolerant.

L. maackii as an invasive species

In the 1950s, *L. maackii* began to escape cultivation in the United States (Luken and Thieret, 1996). Since its initial escape, *L. maackii* has spread to most of eastern and central United States (Figure 1) (EDDMapS, 2020). It has also been documented in parts of Canada (Trisel and Gorchov, 1994).

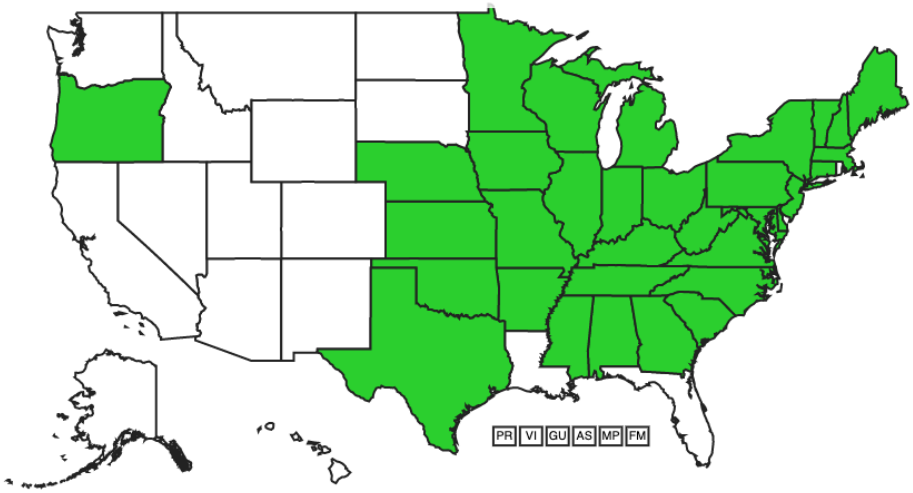

Figure 1. United States map of the geographic distribution of *Lonicera maackii* in February, 2020 (EDDMapS, 2020). *L. maackii* has invaded much of eastern and central United States, and has also spread to parts of Canada (Trisel and Gorchov, 1994).

L. maackii has been able to invade such a wide region for many reasons, including its long-distance seed dispersal, growth pattern, growth season, allelopathic effects on native plants, and resistance to herbivory. McNeish and McEwan detail many of these reasons in their 2016

review. *L. maackii* berry seeds are dispersed far distances by native migratory birds that consume the berries on their southern migration (Ingold and Craycraft, 1983). The rapidly growing, multi-stemmed structure of the *L. maackii* shrub give the plant a dense coverage that shades out other plants very early in its life (Deering and Vankat, 1999). The growing season of *L. maackii* also gives it a competitive advantage over native species. *L. maackii* produces leaves earlier and its leaves last longer into the fall/winter than native plants (McEwan et al., 2009). *L. maackii* also produces chemicals that can have an allelopathic effects on native plants, meaning they negatively affect the growth, survivorship or reproduction of the native plants (Bauer et al., 2012). Finally, *L. maackii* has been shown to be resistant to herbivory by native herbivores compared to native plant species (Lieurance and Cipollini, 2013; Lieurance and Cipollini, 2012).

Invasion of this species has caused many problems, most notably decreased species richness in invaded areas (Collier, Vancat and Hughes 2002; Peebles-Spencer et al., 2017; Hartman and McCarthy 2008, Watling 2011). Collier, Vankat, and Hughes (2002) found lower species richness and abundance under the crowns of *L. maackii* bushes. They also demonstrated that forests invaded by *L. maackii* had decreased species richness for all species and reduced tree seedling density. Similarly, Peebles-Spencer et al. (2017) demonstrated that the presence of *L. maackii* led to decreased species richness, resulting in negative effects directly on plant communities. *L. maackii* invasion has also been shown to cause reduced growth and reproduction of native plant species, affecting native production of fruit, seeds, and flowers (Miller and Gorchov, 2004).

Furthermore, *L. maackii* invasion can negatively affect animals in the invaded areas. In a study by Watling et al. (2011), forest plots invaded by *L. maackii* had lower species richness and evenness of amphibian species when compared to plots without *L. maackii*. The mean maximum

daily temperature was significantly lower in the plots invaded by *L. maackii* than uninvaded plots, indicating that *L. maackii* invasion affects the microclimate. The authors propose that this change in microclimate lead to the changes observed in the amphibian community. This shows that the negative effects of *L. maackii* invasion extend beyond direct plant competitors.

The location of *L. maackii*-invaded areas falls within the paths that migratory birds follow during migration seasons. During fall migration, *L. maackii* berries are accessible to migratory birds as they travel south. However, the nutritional value of *L. maackii*'s fruit has been shown to be inferior to that of native shrubs (Smith, DeSando and Pagano, 2013). *L. maackii* berries have lower fat content and lower energy density than berries of native species. Due to its invasion, *L. maackii* berries are increasingly more available than nutritious native berries, so migratory birds are forced to eat the less nutritious berries. Consumption of these berries also amplifies this problem because of the seed dispersal that occurs when the birds continue on their migration path. Furthermore, because the birds must consume more berries for equal energy value relative to the native berries, the number of *L. maackii* seeds dispersed is even greater.

Beyond their inferior nutritional value, the berries of this species pose an additional threat to migratory songbirds. Recently, songbirds with aberrant red or orange coloring have been observed in Canada and the eastern United States (Hudon and Mulvihill, 2017; Hudon et al., 2013). This pigmentation has been attributed to rhodoxanthin, a deep red keto-carotenoid typically produced by plants, not by birds (Hudon et al., 2013). Hudon et al. (2017) hypothesize that consumption of berries containing rhodoxanthin has caused the unusual coloring. The berries of other invasive bush honeysuckles (*Lonicera* spp.) have been shown to contain rhodoxanthin, so the potential for *L. maackii* berries to have this negative effect on migratory songbirds should be explored (Hudon et al., 2017).

Mechanisms of control

Currently, management of bush honeysuckle does not depend on the species; all species are treated in similar ways. Removal of the honeysuckle plants is the primary goal, but caution must be taken to limit damage to native flora and fauna. There are two primary mechanisms of control: mechanical and chemical (Czarapata, 2005). They may be used independently or in combination. Mechanical management techniques involve physical removal of the shrubs whereas chemical management techniques involve herbicide application (Czarapata, 2005).

When the plants are small or the size of the population is small, the shrubs may be removed using the pull-by-hand technique (Czarapata, 2005). This technique is also useful for large bush honeysuckle plants when herbicides cannot be used. In order for the pull-by-hand technique to be successful, all of the roots must be removed. Otherwise, the shrubs will easily be able to regrow (Czarapata, 2005). Fortunately, bush honeysuckles have shallow root systems, so physical removal of the entire shrub is possible, even for large plants (Smith and Smith, 2010). The pull-by-hand technique works when the soil is wet, such as after a heavy rain (Smith and Smith, 2010).

A common alternative technique to pull-by-hand is cutting or mowing. This technique is common for large communities of honeysuckle when pull-by-hand is not possible (Smith and Smith, 2010). Cutting or mowing removes the above-ground portion of the shrub while leaving the root system intact. If done incorrectly, cutting or mowing can cause more harm than good because sprouting will reoccur from the base, and there will be an increase in the number of stems (Smith and Smith, 2010). To limit regrowth, the root system must be removed or targeted with an herbicide (Smith and Smith, 2010). The cutting or mowing technique is best done during summer because the food reserve is at a low which can reduce the density of regrowth (Smith and Smith, 2010).

If neither the pull-by-hand technique nor the cutting/mowing technique is possible, annual burning may be used as a mechanical removal technique; however, it may not be effective enough on its own (Dolan and Parker, 2002). Burning will kill new seedlings and kill the above-ground portion of larger plants (Czarapata, 2005). A single burn will not effectively control a honeysuckle population; re-sprouting will occur, and the shrubs will regrow with greater density. Burns must be done annually to prevent regrowth (Czarapata, 2005). The burning technique is also limited by the fact that invasion may actually occur more strongly after a burning if the non-native species is not completely eliminated (Zouhar, et al., 2008).

Chemical treatments involve the application of herbicides directly to the honeysuckle bush (Smith and Smith, 2010). Glyphosate, the active ingredient in RoundUp, is an herbicide commonly used for many of the *Lonicera* chemical control strategies (Fuchs and Geiger, 2005; Gorchov, 2005; Smith and Smith, 2010). In a plant, glyphosate is transported with photosynthetic nutrients to the metabolically active areas of the plant. Tissue death occurs within a week of application (Geiger et al., 2005). After application to the stems of bush honeysuckle, glyphosate has been shown to cause structural deformation of the cellular structure of the stem, specifically in the phloem band, indicating tissue death (Fuchs and Geiger, 2005).

Foliar spraying involves spraying the leaves of the honeysuckle bush with herbicide (Geiger et al., 2005). Caution must be taken when using the foliar spraying technique because broad spraying can kill surrounding native species. This technique is best used in the fall because many native species lose their leaves before the honeysuckle, so honeysuckle foliage can be more specifically targeted (Geiger et al., 2005). Common herbicides used for foliar spraying include glyphosate (RoundUp or Accord), 2,4-D + triclopyr (Crossbow), or triclopyr (Garlon 3A, Tahoe 3A) (Smith and Smith, 2010).

Basal spraying is another chemical technique and involves spraying the bottom twelve to eighteen inches of the stem of the honeysuckle bush (Smith and Smith, 2010). This technique works best used during the dormant season (Smith and Smith, 2010). One limitation to this technique is access to the stem. Because of the crowning structure of the honeysuckle, it may be difficult to reach the lower stem with the herbicide (Smith and Smith, 2010).

The cut stump treatment or the cut and paint technique combines mechanical and chemical control techniques (Smith and Smith, 2010; Gorchov, 2005). After the shrub is cut or mowed, an herbicide is applied directly to the stump to target the root system and to prevent re-sprouting (Smith and Smith, 2010). The same herbicides used for foliar spraying may be used for cut stump treatment (Smith and Smith, 2010). One study showed cut and paint to be four times as labor intensive and time consuming as foliar spraying, but foliar spraying was only half as effective for the removal of *L. maackii* (Trisel, 1997).

Selecting an appropriate control mechanism involves the consideration of many factors. In particular, the role of native herbivores may affect the choice of a control mechanism. In a study comparing the basal application and the cut stump treatment techniques to control *L. maackii* as well as the effect of herbivore access, the outcomes of the control measures were dependent on the presence or absence of herbivores (Cipollini et al., 2009). Both chemical control measures led to an increase in species richness in the plots containing the treated plants as compared to untreated controls. In the fenced plots, the cut stump treatment method was more effective than basal spraying, resulting in a higher number of neighboring native plants, neighboring native plants that were taller and produced more fruit, and greater species richness in these plots. However, in the unfenced plots where deer were able to roam, the cut stump technique and the basal spraying

technique produced similar positive effects on the ecosystem. This demonstrates that the appropriate technique depends on other factors beyond the honeysuckle shrubs alone.

Medicinal potential

Lonicera spp. are used in traditional Chinese medicine, and they have also been explored more recently for their potential application to western medicine. The flowers of *Lonicera japonica* are commonly used by traditional Chinese medicinalists to treat colds and fevers (Zhang et al., 2008). Many species of *Lonicera* have been shown to have anti-inflammatory properties, through NO and IL-8 inhibition and suppression of NF-kB and PPAR beta/delta activity (Nikzad-Langerodi et al., 2017). In another study, the *n*-butanol fraction of *L. japonica* extract was shown to have some anti-inflammatory effects against acute, granulomatic, and chronic inflammation, but it was not found to be as effective as other widely used anti-inflammatory agents (Lee et al., 1998).

One of the chemicals found in honeysuckle that is of interest for medicinal purposes is chlorogenic acid, which has been found to have many beneficial effects, including antioxidant, anti-inflammatory, anti-mutagenic effects (Huang et al., 2017; Sasaki et al., 1996; Nakamura et al., 1997). Other components of honeysuckles have also been found to have medicinal properties. *L. japonica* has been used in traditional Chinese medicine to treat influenza (Zhou et al., 2015). In 2006, Ko et al. demonstrated that the use of *L. japonica* led to suppression of viral replication, however they did not determine the specific mechanism of action. In another study, *Lonicera caerulea* was found to have anti-obesity activity and prevent the development fatty-liver in mice (Kim et al., 2018). Mice were fed a high-fat diet, and in mice that were also fed extract from *L. caerulea*, significantly less weight gain and significantly lower incidence of non-alcoholic fatty liver disease were observed.

As demonstrated, some species of *Lonicera* are used in diverse ways for medicinal purposes. However, little research has been done to determine the medicinal properties of *L. maackii* specifically. More work should be done to explore the medicinal potential of *L. maackii.* Investigations can begin by exploring the currently known medicinal properties of other *Lonicera* species. Novel medicinal properties may also be investigated.

Next generation sequencing

Next generation sequencing technologies allow scientists to determine genomic sequences quickly and inexpensively. The first genome sequence was published in 1995 and was the genome of *Haemophilus influenzae Rd.* (Fleischmann et al., 1995). The first plant genome was not sequenced until 2000 when the genome sequence of *Arabidopsis thaliana* was published (Arabidopsis Genome Initiative, 2000). As of 2018, almost 600 complete plant genome sequences had been added to public repositories (Karsch-Mizrachi et al., 2011). Most of the available plant genome sequences that are currently available are for agricultural crops, and there is limited sequencing data available for non-agricultural species (Kersey, 2018). Many chloroplast genome sequences of a range of species have been made publicly available, but whole plant genome sequences are relatively scarce (Nie et al., 2012; Zhang et al., 2014). Recently, more emphasis is being placed on assembling plant genomes. For example, the 10,000-plant genome sequencing project aims to determine the genomic sequence of 10,000 diverse plants to better understand plant genomic variation (Twyford, 2018).

An advantage to using next generation sequencing to understand invasives is that it allows scientists to use a well-established technique to study the underlying molecular mechanisms to determine why invasives are so successful in their non-native environments. For example, the Asian longhorned beetle is an invasive insect that can eat and destroy native tree species. By

sequencing the genome of the Asian longhorned beetle, McKenna et al. (2016) were able to identify genes encoding enzymes that enable the beetle to degrade the natives, including enzymes that degrade polysaccharides in plant cell walls.

Genomics can also help to characterize invasive plant species relative to native plant species, helping to identify plants that are likely to become invasive before an invasion actually takes place. Pysek, et al. found that invasive and native grass species differed in genome size, where species with smaller genomes were more likely to be invasive than specie with larger genomes (2018). In a semi-contradictory study, Pandit, Pocock, and Kunin found that higher ploidy levels were associated with invasive plant species whereas endangered species were more likely to have low ploidy levels (2011). Beyond dissenting information in the literature, there are many gaps in the current knowledge of invasive plants that could be filled with bioinformatics techniques, such as the mechanism of invasion, the genetic basis for the plasticity of invasives, the evolution of the invasive, and intraspecies diversity (Lee, 2002, Ward, Gaskin, and Wilson, 2008). With the study of many invasive plant species, biomarkers to predict invasive plants could be developed, which could help to identify invasive species before a species has escaped cultivation or established a population, ultimately benefiting native species and reducing the economic burden of invasive plants.

Currently, there is no publicly available assembled genomic sequence for any *Lonicera* species. Determining the genomic sequence of *L. maackii* will help to identify key genes that will aid in the understanding of the best control mechanisms, its effect on local bird populations, and its medicinal potential. Having an available genome sequence for *L. maackii* is important to early identification of *L. maackii* in an area so that control measures can be implemented early to prevent the invasive from becoming unmanageable. Genomic evidence could aid in the species

identification of young *L. maackii* plants so that they can be removed before they reach reproductive age and are too large to be easily removed. Furthermore, the development of a bioinformatics pipeline will help scientists who are interested in invasive plant species be able to follow a similar procedure and understand other invasive plant species on the genomic level, thereby providing an entirely new level of information to understand particular species or, when taken together, groups of invasive plant species.

Purpose

The purpose of this study was to assemble and annotate a draft genome of *Lonicera maackii*. Furthermore, the purpose was to provide an analytical pipeline that can be used to assemble genomes of other invasive plant species. A draft genome assembly was built from raw genomic sequencing data, and gene locations were annotated in the genome. The annotated genome was analyzed to identify genes important to *L. maackii*'s success as an invasive, genes that could inform control strategies of *L. maackii* and to investigate the effect of the consumption of the berries by migratory songbirds.

METHODS

The analysis pipeline developed for this study are outlined in figure 2.

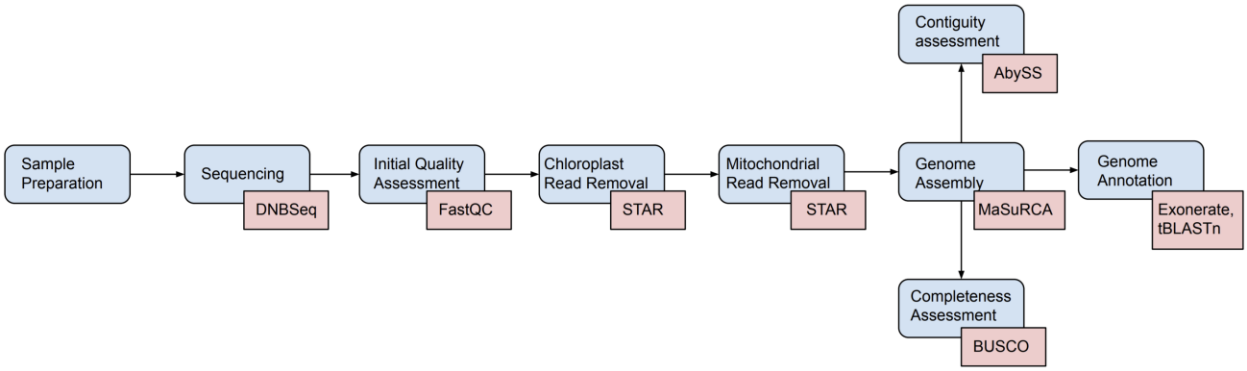

Figure 2. Summary of methods used in this study. Sample preparation was performed for one *Lonicera maackii* individual. Sample was sequenced on DNBSeq platform. Initial quality assessment was performed using FastQC. Chloroplast reads were removed from raw sequencing reads using STAR. Mitochondrial reads were then removed using STAR. The *de novo* genome assembly was then generated using MaSuRCA. Assembly contiguity was assessed using AbySS. Assembly completeness was assessed using BUSCO. Finally, the genome was annotated using Exonerate.

Sample Preparation

Three samples were taken from the leaves of a bush honeysuckle plant a private property in Bristol, New York. Plants on this property with identical morphology were identified as *Lonicera* by a NYS DEC agent. This particular plant was determined to be *Lonicera maackii* based on the shape of the leaves, the berry color, shape, and position on the stem, the characteristics of the bark, flower color, and by analyzing a cross section of the stem.

In the Hudson Lab at RIT, genomic DNA was isolated from the three samples using the Qiagen DNeasy Plant Mini Kit. Samples were ground up and chemically lysed. RNAse was used to remove RNA. The QIAshredder spin column was used to remove cell debris, precipitated

proteins, and polysaccharides and to homogenize the samples. Then, the lysate was added to the DNEasy Plant Mini spin column. DNA bound the column and contaminants were removed during several washes. Finally, DNA was eluted from the column.

Upon arrival to the BGI sequencing facility, the concentration of the DNA samples was determined using a Qubit fluorometer and the DNA BR kit. Sample integrity was determined using agarose gel electrophoresis. A 1% agarose gel was prepared and run at 150V for 40 minutes. Concentration of the three samples were all found to be below the required concentration, so the three samples were pooled into one.

Sequencing

The pooled sample was sequenced using the DNBseq platform at the BGI sequencing facility, using paired-end 100 base pair reads. DNBseq makes use of DNA nanoball technology. The production of DNA nanoballs is based on the replication of circular chromosomes. First, genomic DNA is extracted and fragmented. The ends of the DNA are repaired so that both ends are blunt. Next an adaptor is added to both ends of the DNA, and the DNA fragments are amplified using PCR. Then, the double stranded DNA fragments are separated using heat. The single stranded DNA is ligated to create a circular fragment of DNA, and DNA polymerase continuously replicates the circular DNA fragment many times creating a long piece of single stranded DNA that compacts to form the DNA nanoball. Then, the DNA nanoballs are added to a flow cell and each nanoball is sequenced using fluorescent dNTPs, similar to Illumina sequencing.

Quality control

The initial quality control was performed by the bioinformaticians at BGI prior to distributing the sequencing data. The quality of the data obtained from BGI were assessed using FastQC version 11.9 (Andrews, 2019) (Appendix A-2).

The quality of sequencing data is represented using the Phred quality score system. Phred scores are assigned for each base call and are stored together with the sequence itself. Phred scores are stored as ASCII characters, each corresponding to a particular numeric value. The Phred score measures the quality of the base call. The Phred Q score is related to the probability of an erroneous base call as shown in equation 1 (Shi, Li, and Xu, 2016). A Phred score of 10 indicates that there was a 1 in 10 probability of an incorrect base call or 90% base call accuracy. Similarly, a Phred score of 40 indicates that there was a 1 in 10,000 chance of an incorrect base call or 99.99% base call accuracy.

$$P = 10^{-Q/10} \quad (1)$$

Chloroplast reads were removed from all FASTQ files. The chloroplast genome sequence for *Lonicera maackii* was obtained from GenBank (MN256451.1). Paired end reads were aligned to the chloroplast genome using STAR (Dobin et al., 2013) using the script chloroAlign.sh (Appendix A-4). A maximum of two mismatches were allowed. Unmapped reads were written to new FASTQ files, as specified by the flag --outReadsUnmapped FastX, and were used in the mitochondrial read removal phase.

Mitochondrial reads were removed from FASTQ files generated following chloroplast read removal. The mitochondrial genome sequence for *Helianthus annuus* (common sunflower) was obtained from GenBank (NC_023337.1). At the time of this analysis, this was, to my knowledge, the most closely related mitochondrial genome sequence available. Paired end reads were aligned to the mitochondrial genome using STAR with the script mitoAlign.sh (Appendix A-6). A maximum of two mismatches were allowed. Unmapped reads were written to FASTQ files and were used for *de novo* assembly.

Species identification

A genetic approach was used to corroborate the species identification done by a local DEC agent. The sample from this study was genetically compared to *L. maackii* and *Elaeagnus macrophylla*. At the time of this analysis, there was no reference sequence for the chloroplast genome of *E. umbellata,* so the reference sequence for another member of the *Elaeagnus* genus, *E. macrophylla* was used for genetic comparison.

The reference sequence for the chloroplast genome of *L. maackii* (NC_039636.1) and *Elaeagnus macrophylla* (KP211788.1) were obtained from GenBank. Two pairs of FASTQ files were aligned to each of the reference genomes using STAR (Dobin et al., 2013) allowing for a maximum of two mismatches according to the script find_species.sh (Appendix A-8). The quality of the alignment to each of the two references was analyzed.

In a 2015 study by Choi, Son and Park, the trnH gene was found to be duplicated in Elaeagnaceae, but this gene is not duplicated in the chloroplast reference for *L. maackii.* To determine if the *trnH* gene was duplicated in the sample, three genes that were not known to be duplicated in either species were identified, *trnQ, trnM,* and *trnW.* The sequences of these three genes and of the *trnH* gene were identified in both *L. maackii* and *E. macrophylla*. Using samtools (Li et al., 2009a), the BAM file output from the alignment of one file pair to *L. maackii* chloroplast reference was converted to a FASTA file. A nucleotide BLAST database was made using the alignment output FASTA file (Li et al., 2009b). Using a command line nucleotide BLAST (Camacho et al., 2009), the gene sequences for *trnH, trnQ, trnM, and trnW* were queried against the database generated from the alignment output, according to the script speciesBLAST.sh (Appendix A-11). Only hits with no mismatches, no gaps, and at least 80% of the query sequence present in the alignment were kept. The number of hits for each of the four genes in the filtered hit

list was determined and visualized using R code and the RStudio IDE (R Core Team, 2020; RStudio Team, 2020) using the script trnAnalysis.R (Appendix A-13).

Genome Assembly

FASTQ files output from mitochondrial read removal were used as input files for genome assembly. MaSuRCA (Zimin, 2013) was chosen as the tool for genome assembly because it has been shown to be a high-quality assembler, yielding the highest NG50 scores, longer sums of contig lengths, highest percentages of BUSCO reference genes, and highest scaffold statistics in a comparison of genome assemblers (Olsen, 2019). No additional pre-processing of the reads was performed after chloroplast and mitochondrial read removal.

The sr_config.txt file was written according to MaSuRCA documentation (Appendix A-15). In the "DATA" section, each file pair was marked as paired end with the notation "PE=", was given an alphabetical prefix, and was indicated to have paired-end length of 200 bp. One file pair was not used in the assembly because MaSuRCA was not able to properly process the file. The parameter EXTEND_JUMP_READS was set to "0" because the MaSuRCA documentation recommends setting to "1" only for Illumina jumping library reads shorter than 100bp. The GRAPH_KMER_SIZE parameter, which is the k-mer size for de Bruijn graph, was set to "auto," which allowed MaSuRCA to use the read data and GC content to determine the optimal k-mer size. The parameter USE_LINKING_MATES was set to "1", as recommended for Illumina-only assemblies. Because the documentation did not indicate any parameter recommendations for reads sequenced on the BGI-platform, this parameter was set to 1 based on the assumption that the reads from BGI-seq would be similar to those sequenced on an Illumina platform. The parameter LIMIT_JUMP_COVERAGE was set to "300" according to the documentation recommendation to set this parameter to 60 for bacteria and 300 for all other organisms. For the Celera Assembler

parameter, CA_PARAMETERS = cgwErrorRate was set to 0.15. The documentation recommends this parameter be set to 0.25 for bacteria and between 0.1 and 0.15 for all other organisms. The JELLYFISH_HASH_SIZE was set to 60,000,000,000. To indicate that the SOAPdenovo assembly should not be used, the parameter SOAP_ASSEMBLY was set to "0." Finally, eight CPU threads were used to run the assembly, as indicated by the parameter NUM_THREADS = 8.

The assembly script was generated from the configuration file using the command:

```
/usr/local/bin/MaSuRCA/bin/masurca sr_config.txt
```

This generated the file assembly.sh which was then executed to run the assembly.

Assembly Evaluation

In order to evaluate the quality of the assembly, both the contiguity and completeness were evaluated. To evaluate the contiguity, the function abyss-fac from the command line tool ABySS was used (Simpson et al., 2009), according to the script get_contig_stats.sh (Appendix A-20). This program requires an estimated genome size to calculate the contiguity statistics for an assembly. A direct genome size estimate for *L. maackii* was not available in the literature, so it was estimated using information from other *Lonicera* species. Wang and Wang (2005) found that both *L. japonica* and *L. maackii* are diploid and both have a chromosome count of 2n = 18. According to a study by Chen et al. (2017) the 1Cx value for *L. japonica* was found to be 1,135 Mbp. The 1Cx value is the DNA content of the haploid chromosome set with chromosome number x (Chen et al., 2017). The 1Cx value for *L. maackii* was estimated to be approximately the same as *L. japonica* because both are diploid and have 2n = 18. Therefore, a value of 1,135 Mbp was used for the 1Cx value for *L. maackii*. Multiplying the 1Cx value by the ploidy level, two, the total estimated genome size was 2,270 Mbp. This genome size estimate was used to calculate the contiguity statistics using abyss-fac.

To evaluate the completeness of the assembly, BUSCO was run with the eudicot_odb10 lineage using the script runBUSCO.sh (Simão et al., 2015) (Appendix A-22).

Genome Annotation

Annotation using Exonerate

To identify gene locations within the assembled genome, Exonerate (Slater and Birney, 2005), a tool for pairwise alignment, was used to annotate the assembled contigs with the proteome of *Helianthus annuus* (UniProt proteome ID UP000215914), the most closely related species with a reference proteome. The code to run exonerate is available in the bash script runExonerate.sh (Appendix A-24).

The protein2genome model was used to perform pairwise alignment between the *H. annuus* proteome as the query and the assembled contigs as the target. This alignment model considers gaps and frameshifts when performing the alignment, allowing for the prediction of gene location, coding regions, introns, and exon boundaries. The --showtargetgff flag was used to convert the alignments to GFF format. The flag --showalignment was set to 'no' to reduce the size of the output file. The parameter --fsmmemory was set to 500 to supply 0.5 Gb of memory for the finite state machine's heuristic analyses. The parameter --seedrepeat was set to 100, which required that 100 seeds be found before extension could occur. The GFF sections of the Exonerate output file were then extracted and written to another file, lonicera_maackii_exonerate_tmp.gff. Then, the R script exonerate2gff.R was used to provide more information in the "attribute" column of the GFF file, specifically the species used for annotation, the gene symbol, the protein name, and the UniProt accession number for the protein (Appendix A-32). The final annotation generated with exonerate was written to the file lonicera_maackii_exonerate.gff

Annotation with tBLASTn

Because of computational challenges associated with using exonerate, a second method of annotation was implemented using a command line translated BLAST search (Camacho et al., 2009). Gene locations in the assembled genome were predicted by again using the proteomes of *Helianthus annuus* (UniProt proteome ID UP000215914) and *Arabidopsis thaliana* (UniProt proteome ID UP000006548). A nucleotide BLAST database was generated from the assembled scaffolds. Then, the reference proteomes were queried against the scaffold nucleotide BLAST database. Default scoring parameters were used for tBLASTn. The BLOSUM62 scoring matrix was used with a gap opening cost of 11 and a gap extension cost of 1. Output was generated in a tab-delimited file, as specified by the command line option -outfmt 6. The following information was gathered for each BLAST hit: the query sequence id (qseqid), the subject sequence id (sseqid), the percent of identical matches (pident), the number of identical matches (nident), the length of the alignment (length), the length of the subject (slen), the length of the query (qlen), the number of mismatches (mismatch), the total number of gaps (gaps), the expect value (evalue), the bit score (bitscore), the location of the start and end of the alignment in the subject (sstart and send respectively), and the raw score (score). The script annotationBLAST.sh was used (appendix A-26).

In order to determine adequate threshold to use for filtering the BLAST hits, a comparative translated BLAST search was performed using a draft assembly of *Crucihimalaya himalaica*, a relative of the very well-studied plant *Arabidopsis thaliana*. The draft assembly was compared to the reference proteome for *A. thaliana* (UniProt proteome ID: UP000006548) using the same procedure described for *L. maacki*. The assembled contigs were obtained from NCBI (BioProject PRJNA521295; GenBank accession number GCA_004349715.1). Then, a nucleotide BLAST

database was generated from the contigs. Finally, a translated BLAST search was performed by querying the Arabidopsis proteome against the contig nucleotide BLAST database. The distribution of alignment lengths and e-values for the two translated BLAST searches were compared. Minimum alignment lengths and maximum e-values were identified to define valid BLAST hits using the distribution comparison and previous thresholds identified in the literature (Mochida et al., 2016, Zhang et al., 2019, Kumari, Singh, and Rai, 2020, and Peng et al., 2014).

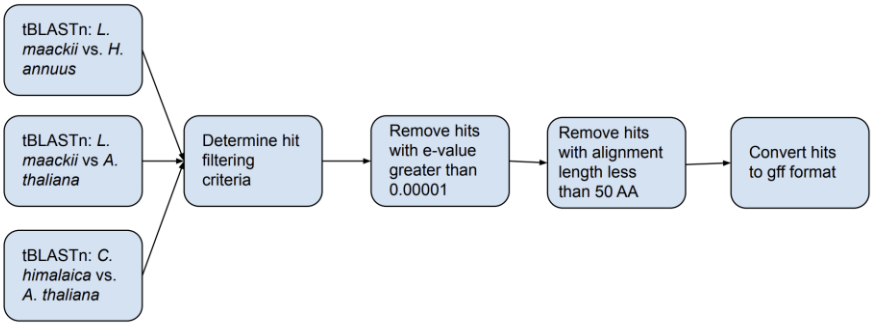

Figure 3. Steps to annotating *L. maackii* genome using the tBLASTn method. Three command line translated BLASTs were run with the following query-subject pairs: *H. annuus* proteome-*L. maackii* genome assembly, *A. thaliana* proteome-*L. maackii* genome assembly, and *A. thaliana* proteome-*C. himalaica* genome assembly. Then, criteria were determined to remove hits that were likely false positives (high e-value) or had short alignment lengths. Hits with e-value greater than 0.00001 and alignment length less than 50 amino acids were removed. Finally, remaining hits were converted to GFF format.

The raw BLAST output was converted to a GFF format using a multi-step process, as described in the script blast2gff.R (Appendix A-28). Hits with an e-value of greater than $1x10^{-5}$ or an alignment length less than 50 amino acids were removed from the BLAST output generated from alignment to *H. annuus* proteome and *A. thaliana* proteome. Then, for each of the proteins in the proteomes, the hit with the lowest e-value was kept. If more there were multiple hits with

the lowest e-value for a given protein, both were kept. Finally, the BLAST output fields were mapped to the corresponding GFF fields and the results were written to the annotation file lonicera_maackii_blast.gff

The R script compare_annotations.R was used to summarize and compare the GFF files created by the two different methods (Appendix A-34).

RESULTS

BGI sequencing produced high quality data.

Three DNA samples from one individual were sent to the BGI sequencing facility (Table 2) (BGI Communications, 2019). Sample 1 had a DNA concentration of 4.6 ng/uL, sample 2 had a concentration of 5 ng/uL, and sample 3 had a concentration of 5.8 ng/uL. Each of the three samples had a volume of 74 uL. Sample 1 had a total DNA mass of 0.34 micrograms, sample 2 had a total DNA mass of 0.37 micrograms, and sample 3 had a total DNA mass of 0.43 micrograms. Because the total mass was less than one microgram in all three samples, the three samples were pooled to make one sample. The DNA in all three of the samples was found to be slightly degraded (Figure 4) (BGI Communications, 2019).

Table 2. Sample information from BGI sequencing facility[1]

No.	Sample Name	Sample Number	Tube No.	Concentration (ng/μL)	Volume (μL)	Total Mass (μg)	Sample Integrity	Library Type	Test Result	Remark
1	PL2-1	8521910012297	1	4.6	74	0.34	Degraded Slightly	≤800bp Insert Size	Unqualified	c<1.5ng/μl,m<1μg, proposed to resend the sample
2	PL2-2	8521910012298	1	5	74	0.37	Degraded slightly	≤800bp Insert Size	Unqualified	c<1.5ng/μl,m<1μg, proposed to resend the sample
3	PL2-3	8521910012299	1	5.8	74	0.43	Degraded slightly	≤800bp Insert Size	Unqualified	c<1.5ng/μl,m<1μg, proposed to resend the sample

[1]Adapted from BGI Communications, 2019

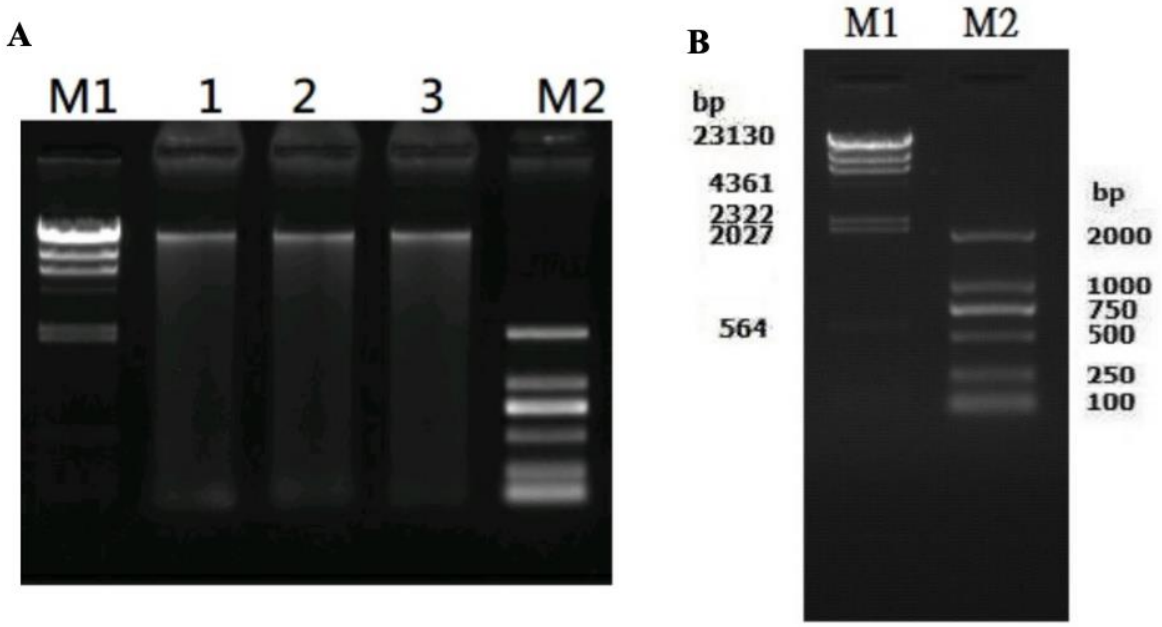

Figure 4. A) Agarose gel electrophoresis of the three DNA samples, 1, 2, and 3. M1 and M2 represent DNA ladders. The gel shows some degradation of all three of the DNA samples. **B)** Gel showing the reference sizes of the two DNA ladders, M1 and M2. From BGI Communications, 2019.

The pooled DNA sample was sequenced on BGI's DNB-seq platform. The initial quality of the pooled sequencing data obtained from BGI is shown in Figure 5 (BGI Communications, 2019). Paired end reads of length 200 base pairs total are shown, where each read is 100 base pairs and is concatenated with its matching reverse read. Approximately 19 percent of the bases in the entire sample were cytosine, approximately 19 percent were guanine, approximately 31 percent were adenine, and approximately 31 percent were thymine (Figure 5A). Many of the sequenced bases were high quality, with most of the Phred quality scores in the range of 35 to 37 (Figure 5B).

The quality of the data obtained from BGI was analyzed in more detail using the program FastQC, which provides information about the sequences in each of the FASTQ files. Sample output from FastQC is shown in figures 6 through 10. Figures represent the output for a single FASTQ file, but they are representative of the output from the other FASTQ files. Figure 6 shows the distribution of quality scores at each position along the reads in the file, where the yellow box of each boxplot represents the interquartile range of the data, the upper whisker represents the 90^{th} percentile, and the lower whisker represents the 10^{th} percentile. The green-shaded region represents very good quality calls, the orange-shaded region represents reasonable quality calls, and the red-shaded region represents poor quality calls (Andrews, 2010). Almost all the data fell into the green region, indicating that the base calls are high quality calls.

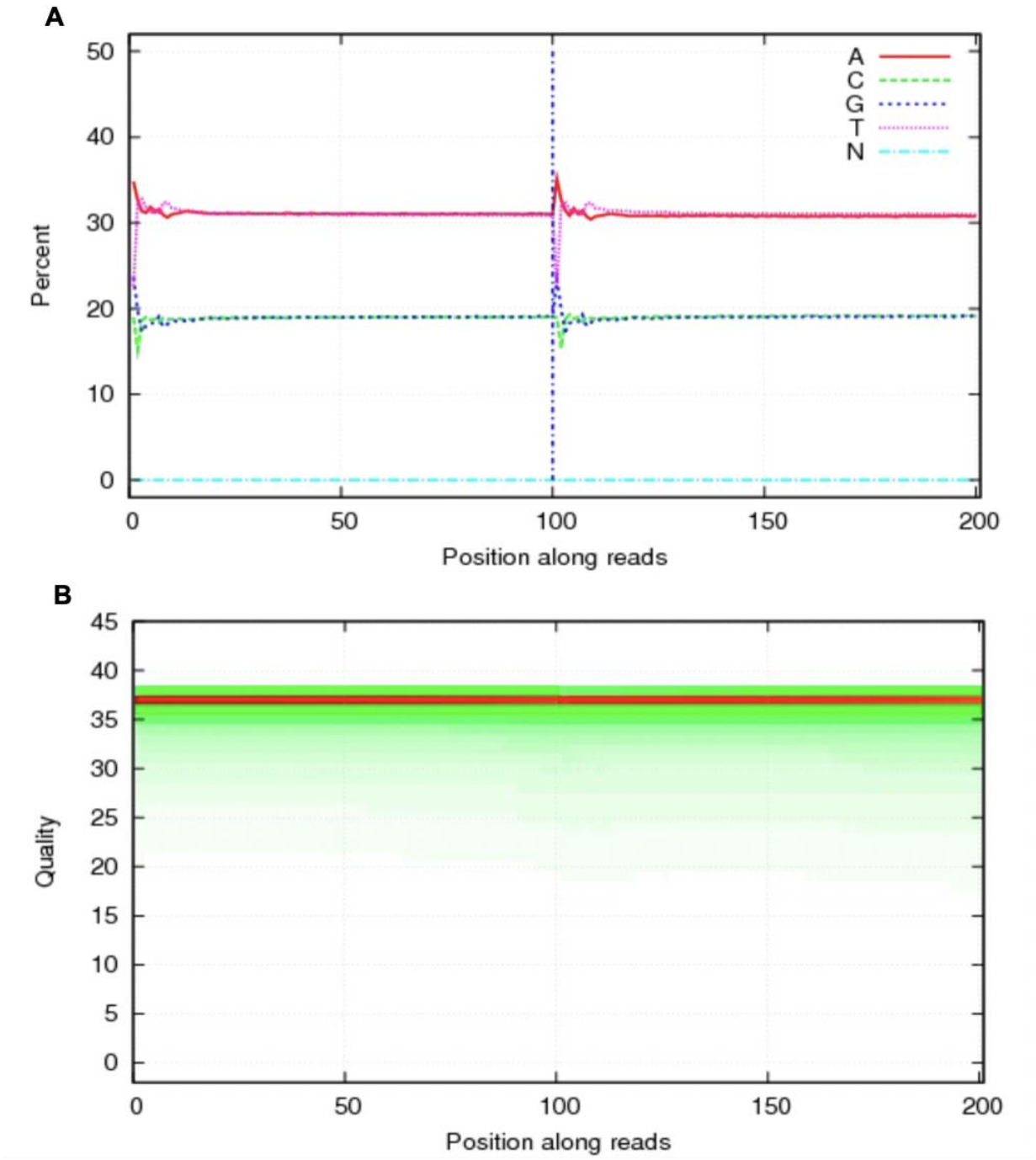

Figure 5. A) Base percentage composition along the length of the reads after initial quality control performed by BGI. **B)** Distribution of quality scores along the length of the reads after initial quality control performed by BGI. From BGI Communications, 2019.

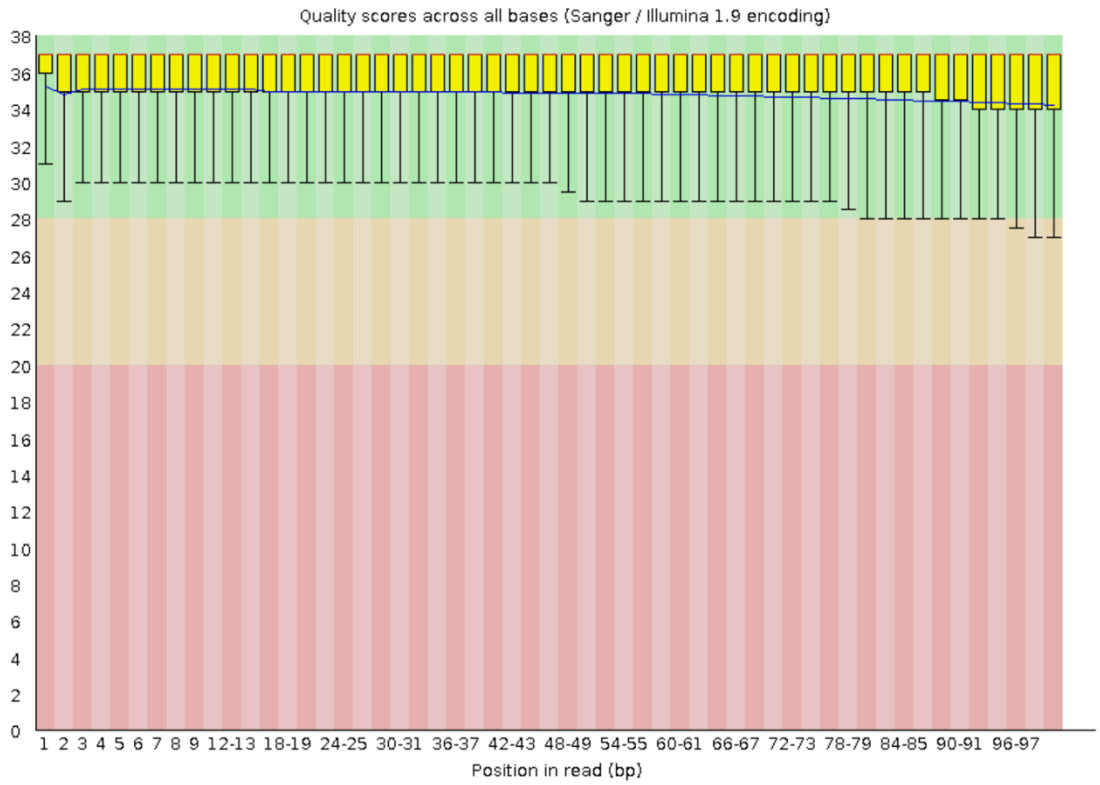

Figure 6. Sample FastQC output of per base sequence quality. The raw reads obtained from BGI had relatively high quality scores per base. All fastq files showed similar per base sequence quality.

Figure 7 shows the distribution of the average base call quality per read. The purpose of this quality visualization is to determine if there are reads with low quality overall. The average quality per read was a Phred score of 36. There was no spike indicating a large number of sequences with low average quality. Taken together, this information suggests that most of the reads were of high average quality.

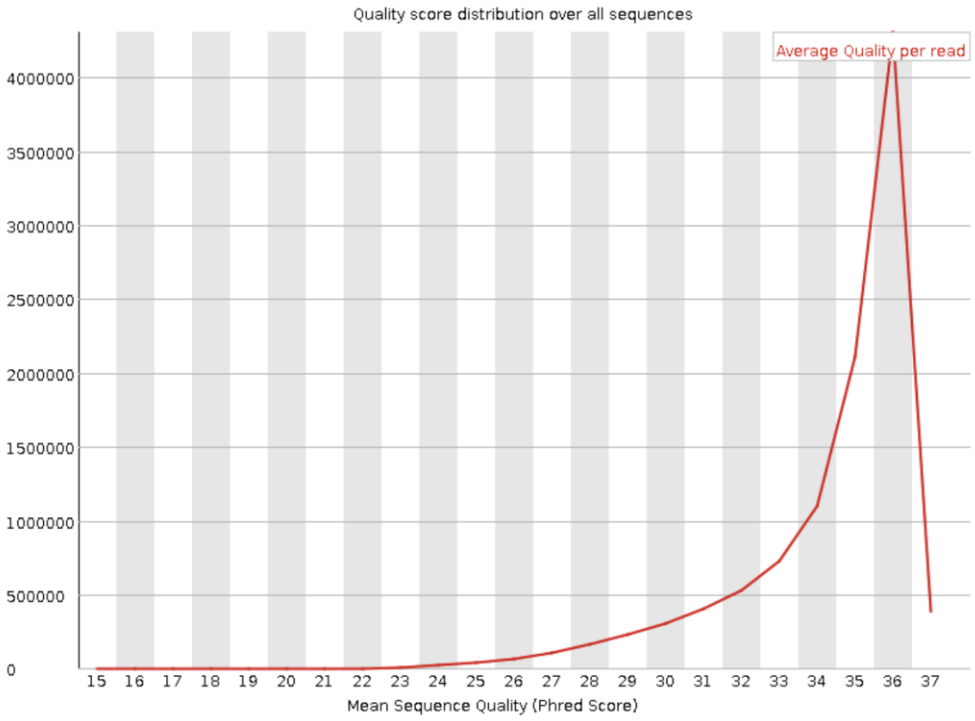

Figure 7. Sample FastQC output for the distribution of average sequence quality scores. Many of the reads had a mean sequence quality score greater than 30, suggesting that the raw data from BGI were high quality. The tail of the distribution stretches into quality scores below 30, but the majority of reads were of high average quality. All FASTQ files form BGI showed similar mean sequence quality distributions.

Figure 8 shows the sequence content across all bases in the reads. At the beginning of the reads, some unevenness in percent base calls was observed, but base call percentages leveled out after 10 base pairs. At the plateau, approximately 19 percent of bases were guanine, 18 percent were cytosine, 32 percent were adenine, and 31 percent were thymine. These sequence content percentages triggered a warning from the FastQC program because the difference between adenine and thymine was greater than 10% at position one, where thymine content was approximately 23 percent, adenine content was approximately 45%, cytosine content was approximately 18 percent,

and guanine content was approximately 24 percent. This warning occurred in the FastQC output for all of the FASTQ files.

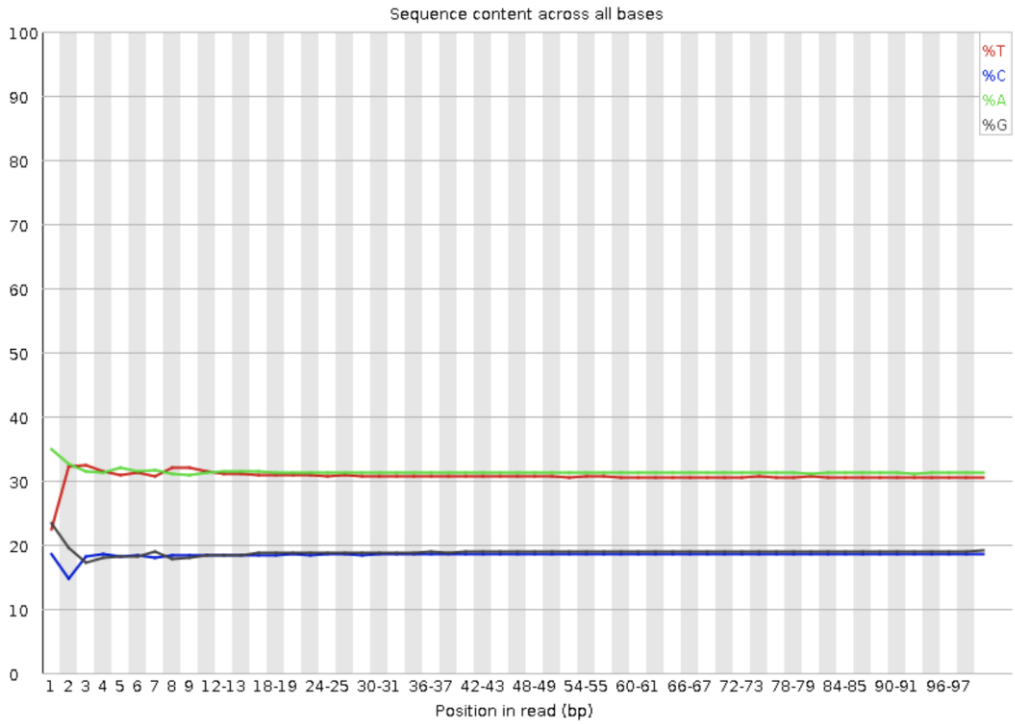

Figure 8. Sample FastQC output for per base sequence quality. Sequence content was approximately constant across the length of the reads. Approximately 19% of bases were G, 18% were C, 32% were A and 31% were T. All FASTQ files had similar sequence content.

Figure 9 shows the distribution of mean GC content per read. The theoretical distribution represents a normal distribution centered at the overall genome GC content (modeled as the mode of the GC content of the observed data). The observed distribution of mean GC content per read very closely matched the theoretical distribution, further corroborating the claim that the data are very high quality.

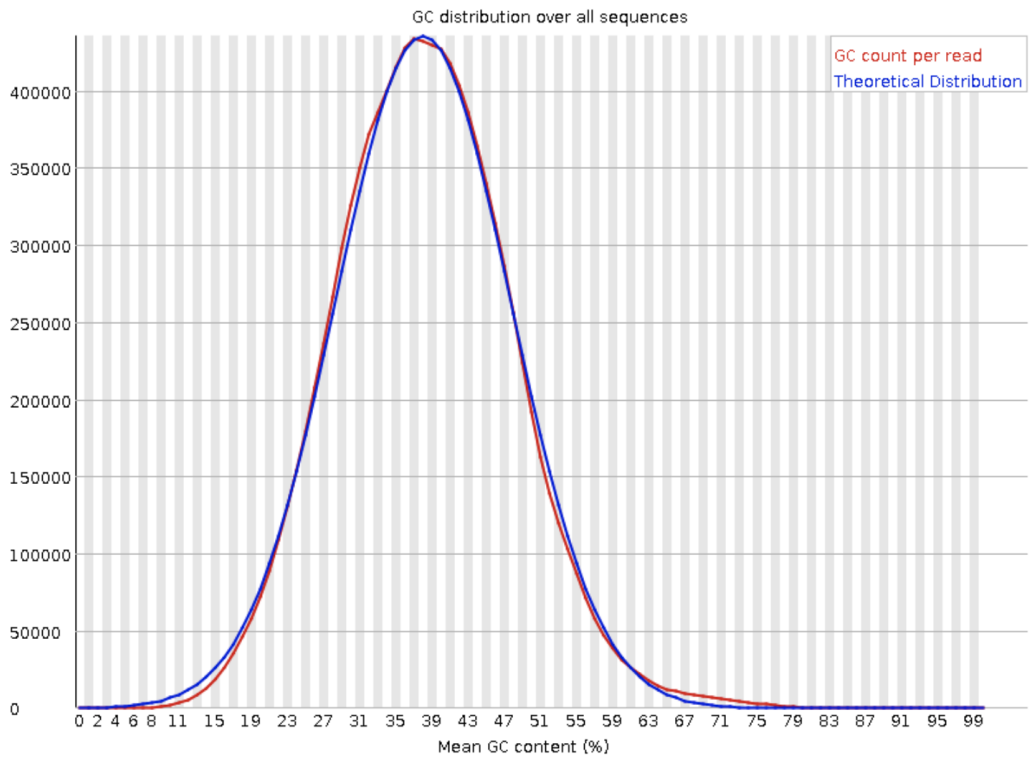

Figure 9. Sample FastQC output for per sequence GC content. The distribution for GC count per read closely matched the distribution for the theoretical GC content, suggesting that the reads were of high quality. All FASTQ files showed similar per sequence GC content with the GC count distribution closely matching the theoretical distribution.

Figure 10 represents the level of duplication of sequences in the FASTQ file. The blue line represents the percentage of sequences for a given duplication level bin from the original data, and the red line represents the percentage of sequences for a given duplication level bin if the sequences were deduplicated. A peak in sequence duplication was observed for the bin >10, which could be attributed to repetitive regions that very commonly occur in plants.

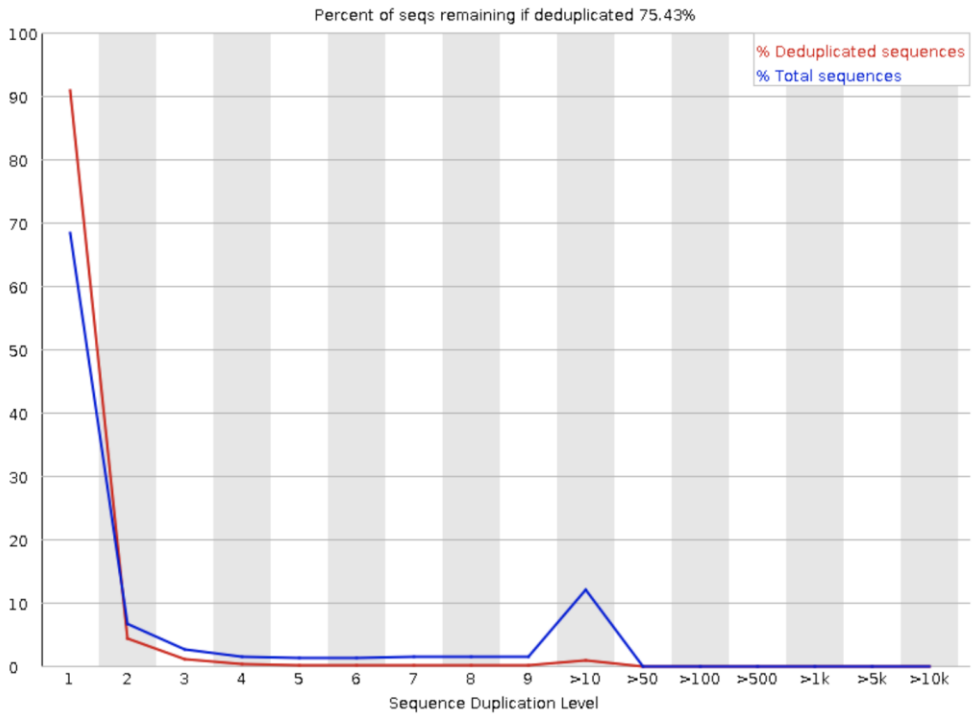

Figure 10. Sample output for sequence duplication levels. There was one peak in duplicated sequences at >10 percent in the output for each of the FASTQ files, but no particular sequences were flagged as being overrepresented.

Phenotypic and genetic evidence support species identification as *L. maackii.*

Before proceeding with the assembly, phenotypic and genetic analyses were performed to corroborate the species identification performed by the NYS DEC officer. Photos of the sampled individual are shown in Figure 11 (Osier personal communications, 2020). The phenotype of the sampled individual was compared to reference images for the four bush honeysuckles (Table 1) and to *Elaeagnus umbellate* (commonly, autumn olive), another invasive plant found in NYS (WNY PRISM, 2021).

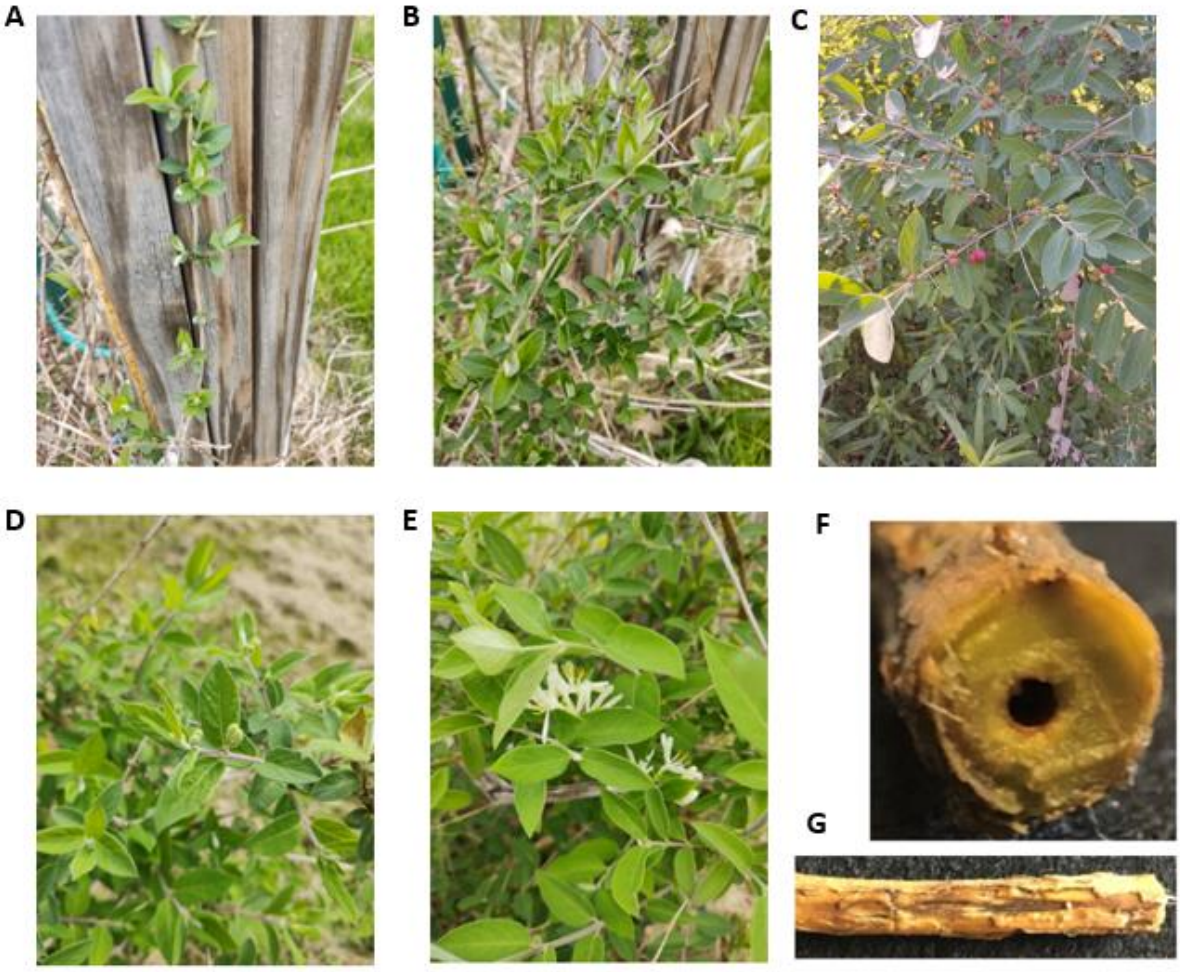

Figure 11. Phenotypic identification of the plant used for sequencing. **A)** Stem. **B)** Multiple stems. **C)** Berries. **D)** Flower buds. **E)** Flowers. **F)** Stem cross section. **G)** Bark. The leaves are opposite and the flowers appear in pairs. Taken from Osier personal communications, 2020.

The sampled individual had leaves that were opposite each other, elliptical in shape and with a long narrow point. The flowers were white and occurred in pairs. The berries were red and round. All of these characteristics support the identity of the individual as *L. maackii*.

Genetic analysis further supported the phenotypic finding that the sampled individual was not *E. umbellata.* Two genetic techniques were used, chloroplast alignment and select *trn* gene quantification. Chloroplast alignment was performed using two chloroplast reference sequences,

one for *L. maackii* and one for *Elaeagnus macrophylla*. Due to computational limitations, only two pairs of fastq files were used for the chloroplast alignment. The percentage of reads that mapped to each of the two reference sequences is shown in Table 3. In the first file pair, zero percent of reads mapped uniquely to both reference sequences. In the same file pair, 1.14% of reads mapped to multiple loci in the *E. macrophylla* chloroplast genome and 3.07% mapped to multiple loci in the *L. maackii* chloroplast genome. The low mapping rate suggests that not many chloroplast sequences were present in this first file pair. In the second file pair, more reads mapped uniquely to the chloroplast genome of *L. maackii* than to that of *E. macrophylla*. Approximately 15.57 percent of reads mapped uniquely to the *L. maackii* chloroplast genome and 0.85% of reads mapped uniquely to the *E. macrophylla* chloroplast genome.

Table 3. STAR alignment results for species identification.

		L. maackii	***E. macrophylla***
File pair 1	Mapped uniquely	0%	0%
	Mapped to multiple loci	3.07%	1.14%
File pair 2	Mapped uniquely	15.57%	0.85%
	Mapped to multiple loci	6.29%	1.54%

Because of the low unique mapping rate of the reads in the first file, only the alignment of the second file pair was used for the BLAST quantification of the selected *trn* genes. Fewer hits were observed for the *trnH* gene than the other three genes (Figure 12). If the *trnH* gene were duplicated, we would expect to see approximately twice as many hits compared to the non-duplicated *trn* genes, *trnM, trnQ,* and *trnW*. This analysis did not provide evidence for duplication of the *trnH* gene in the sampled individual. These results supported the findings from the chloroplast alignment that the sampled individual was not *E. umbellata.*

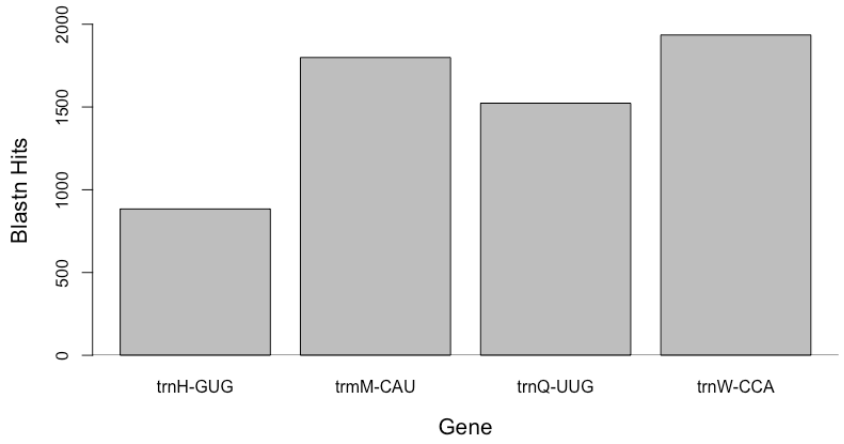

Figure 12. Quantification of *trnH, trnM, trnQ,* and *trnW* genes in the chloroplast reads of the sequenced sample. BLASTn hits were only considered in the quantification process if there were no mismatches, no gaps, and at least 80% of the query sequence was present in the alignment. The data do not support the *trnH* gene being duplicated in the sequenced sample.

Chloroplast and mitochondrial DNA was sequenced but removed.

Typical NGS pipelines involve several quality control steps, such as low quality read removal and trimming the low quality ends of reads, to remove low quality data before assembly or alignment to increase the quality of the end result. However, MaSuRCA has built in quality control procedures and has been found to perform better when raw data is input to MaSuRCA's pipeline. Therefore, no traditional quality control procedures were performed on the data. However, reads likely corresponding to the chloroplast and mitochondria were removed before MaSuRCA (Table 3). Raw reads were first aligned to the reference for the chloroplast of *L. maackii*. The average number of input reads across all the FASTQ files was 11816012.675 reads (σ^2 = 1357193.490 reads). Across all input files, an average of 14.898 percent of reads (σ^2 = 2.491%) uniquely mapped to the chloroplast reference genome. An average of 78.786 percent of reads (σ^2 = 2.945%) per fastq file did not map to the chloroplast reference. These unmapped reads

were then used as the input for alignment to the mitochondrial genome of *H. annuus*. The average number of reads in the FASTQ files input to mitochondrial alignment was 9294399.050 reads (σ^2 = 980834.640 reads). On average, only 0.704 percent of reads (σ^2 = 0.116%) uniquely mapped to the mitochondrial reference, and 99.163 percent of reads (σ^2 = 0.133%) were unmapped. These unmapped reads, which did not align to either the chloroplast or mitochondrial genomes, were used as input for assembly with MaSuRCA.

Table 4. Chloroplast and mitochondrial read alignment.

	Chloroplast alignment[1]	Mitochondrial alignment[1]
Number of input reads	11816012.675 ± 1357193.490	9294399.050 ± 980834.640
Percent uniquely mapped reads	14.898 ± 2.491	0.704 ± 0.116
Percent of unmapped reads	78.786 ± 2.945	99.183 ± 0.133

[1]mean ± standard deviation

***De novo* assembly with MaSuRCA yielded a contiguous and complete genome sequence.**

Two FASTA files are generated by MaSuRCA, genome.ctg.fasta and genome.scf.fasta. The file genome.ctg.fasta contains the assembled contigs, and the genome.scf.fasta file contains the scaffolds. For MaSuRCA, the final assembly output is written to the genome.scf.fasta file. A scaffold is an assembly of some contigs, using mate-pair information (from paired end reads). Therefore, the assembled fragments in the scaffold file are longer, resulting in fewer total fragments and a more contiguous assembly (Table 5).

A total of 205,041 scaffolds were assembled from the input reads (Table 5). The minimum scaffold length was 103 base pairs, and the maximum scaffold length was 161,169 base pairs. The sum of the lengths of the scaffolds was 792.6 million base pairs. The estimated genome size was 2,270 million base pairs.

Table 5. Contiguity statistics for contig and scaffold assembly files.

Contiguity Statistic	genome.ctg.fasta	genome.scf.fasta
Number of contigs	234,255	205,041
n:1	234,255	205,041
L50	19,087	12,727
LG50	234,255	205,041
NG50	86	103
N80	2,338	2,729
N50	10,564	15,582
N20	25,619	39,209
E-size	15,282	22,482
Maximum contig length	120,896	161,169
Minimum contig length	86	103
Sum of contig lengths	792.6e6	792.6e6

The N50 was 15,582 base pairs, meaning when summing the lengths of the scaffolds from largest to smallest, the scaffold that made the sum 50 percent of the total assembled genome size was 15,582 base pairs. A larger N50 indicates a more contiguous genome assembly. The NG50 was 103 base pairs. The NG50 is similar to the N50, but the NG50 uses the estimated genome size instead of the total assembled genome size. Therefore, when summing the scaffolds from largest to smallest, a scaffold of length 103 base pairs made the summed length half the estimated genome size. The estimated genome size was much larger than the assembled genome size (sum of the scaffolds), causing the NG50 to be much smaller than the N50. The L50 was 12,727, meaning 12,727 scaffolds were as long as or longer than the N50 length of 15,582 base pairs. The LG50 was 205,401, meaning 205,401 scaffolds were as long as or longer than the NG50 length of 103 base pairs. A smaller L50 or LG50 suggests a more contiguous genome assembly because fewer scaffolds were needed to achieve half the genome size, assembled and estimated respectively, thereby indicating that the scaffolds were relatively long.

The completeness of the assembled genome was evaluated using BUSCO (eudicot_odb10 lineage), which revealed that the genome was relatively complete (C:86.0% [S:81.4%, D:4.6%], F:6.6%, M:7.4%, n:2326). Only 200 out of 2,326 BUSCOs were missing from the assembled genome. While the contiguity statistics that incorporate estimated genome size suggest that the assembly was not complete, the high percentage of complete BUSCOs and low percentage of missing BUSCOs suggests that the genome assembly was complete, thereby indicating that the actual genome size is far smaller than the genome size estimated based on the size of related species of *Lonicera.*

BLAST provides more annotations than exonerate under computational restrictions

BLAST hits were filtered based on alignment length and on the e-value. To determine adequate thresholds for each, the distributions of the alignment length and e-value were plotted to look for a natural breaking point. A natural break was not observed for e-value (Figure 13), but a natural break was observed at an alignment length of 50 amino acids (Figure 14). An e-value threshold of $1x10^{-5}$ was chosen. The e-value is a measure of the probability that the hit was found by chance. With an e-value threshold of $1x10^{-5}$, there is a one in 10,000 chance that the hit was due to chance. The distribution of alignment lengths and e-values for the two BLAST runs with *L. maackii* closely matched those of the BLAST run for *C. himalaica.*

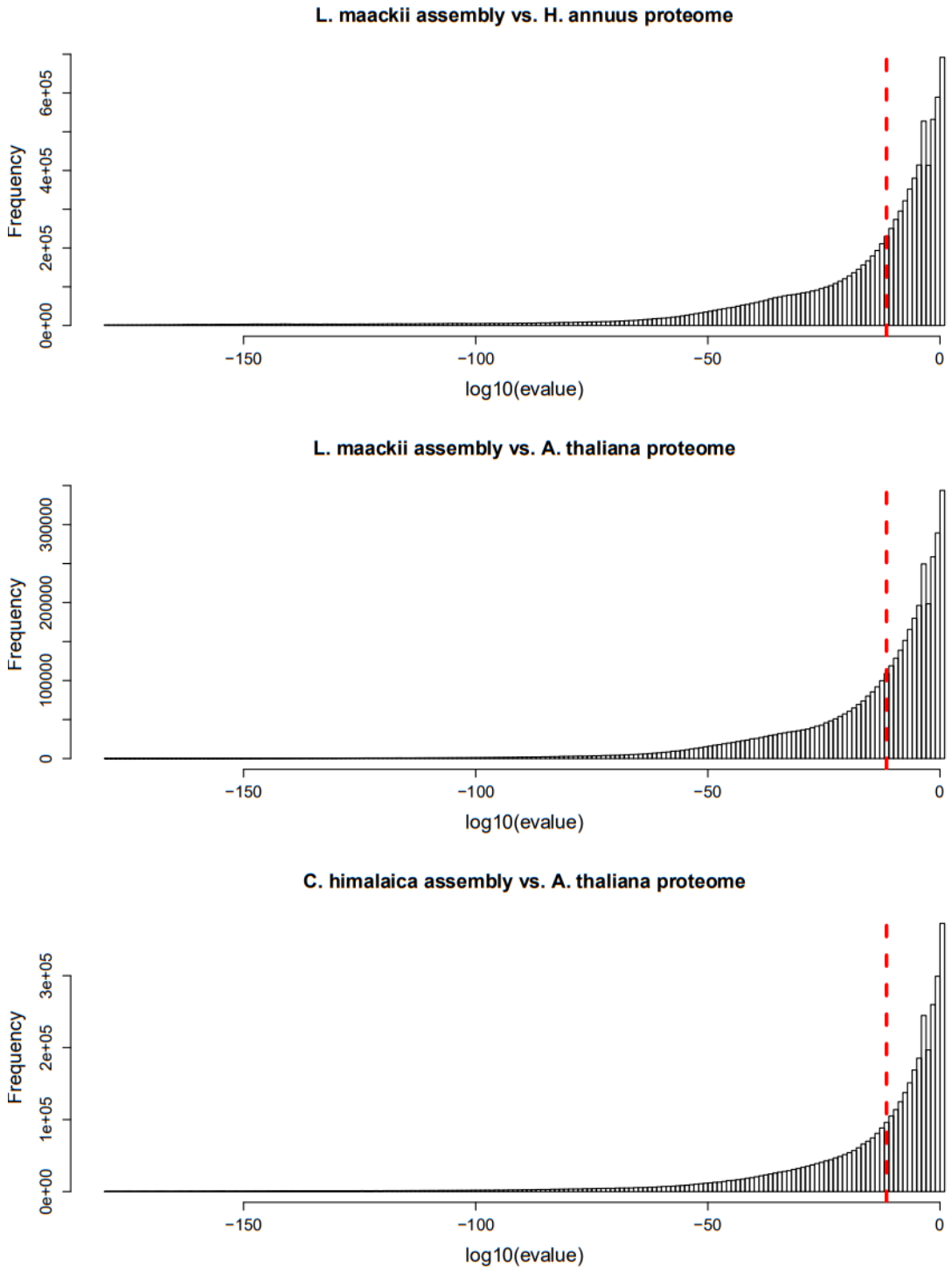

Figure 13. Distribution of alignment e-values for the tBLASTn for *L. maackii* vs. the proteome of *H. annuus*, *L. maackii* vs. the proteome of *A. thaliana*, and *C. himalaica* vs. the proteome of *A. thaliana.* The red dashed line represents the e-value threshold of $1x10^{-5}$.

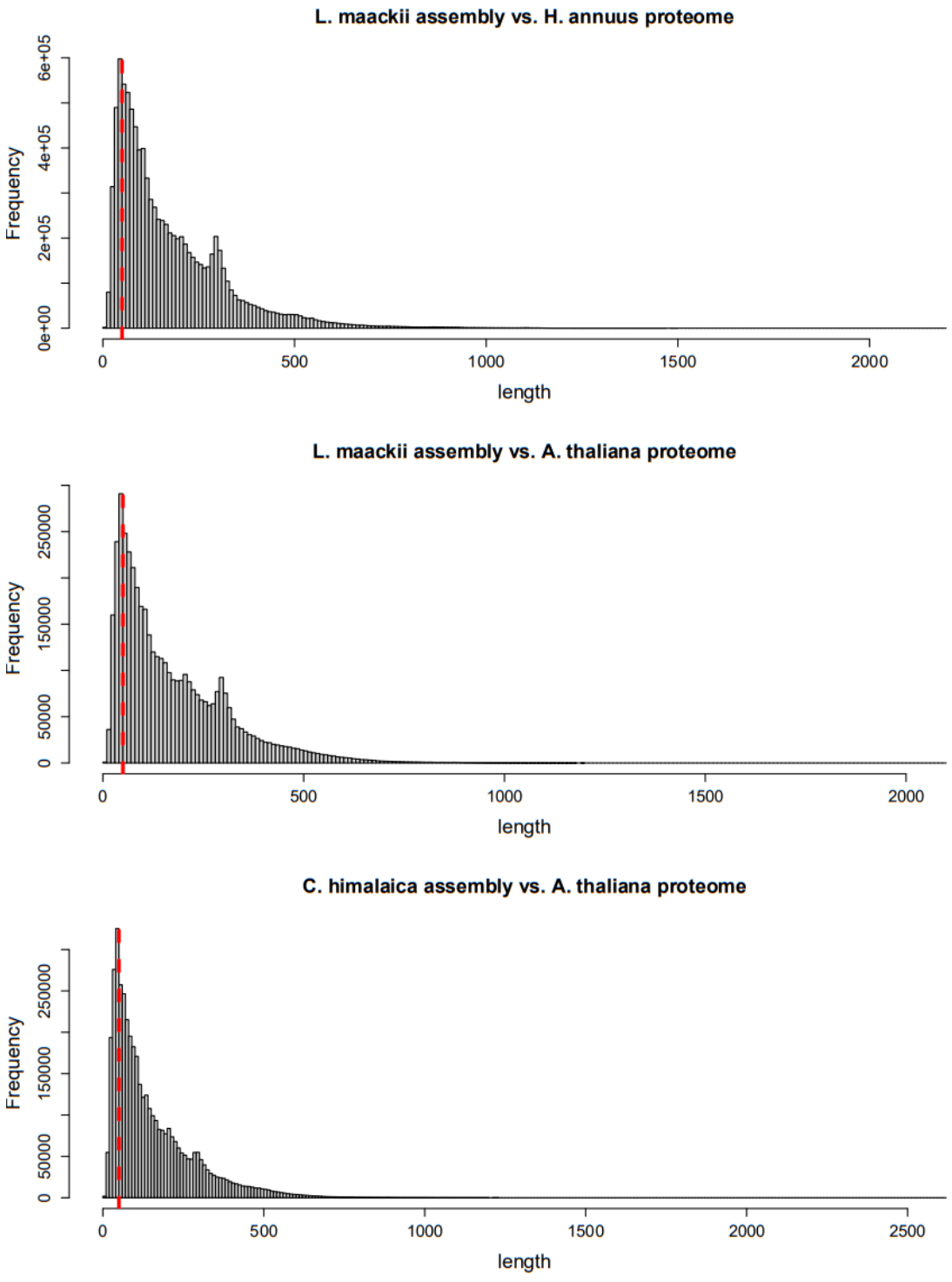

Figure 14. Distribution of alignment lengths for the tBLASTn for *L. maackii* vs. the proteome of *H. annuus*, *L. maackii* vs. the proteome of *A. thaliana*, and *C. himalaica* vs. the proteome of *A. thaliana.* The dashed red line represents the alignment length threshold of 50.

After filtering the BLAST results, the BLAST method of annotation led to finding 22,884 of the 27,468 *A. thaliana* genes and 39,827 of the 51,240 *H. annuus* genes in the *L. maackii* genome assembly. By the exonerate method of annotation, only 3,913 of the 51,240 *H. annuus* genes were found in the *L. maackii* genome assembly. There were 35,917 genes from the *H. annuus* proteome that were annotated by BLAST but not by exonerate. There were only three *H. annuus* genes that were found by exonerate but not by BLAST. These genes were HannXRQ_Chr17g0547701 (putative sodium/solute symporter, accession number A0A251RTG8), HannXRQ_Chr11g0321301 (uncharacterized protein, accession number A0A251T688), and HannXRQ_Chr11g0321291 (putative extension domain-containing protein, accession number A0A251T646). By the Wilcoxon rank sum test, exonerate produced significantly longer alignments than tBLASTn across all BLAST annotations and exonerate gene annotations for the *H. annuus* genes ($p < 0.001$, $W = 1487200000$; Figure 15).

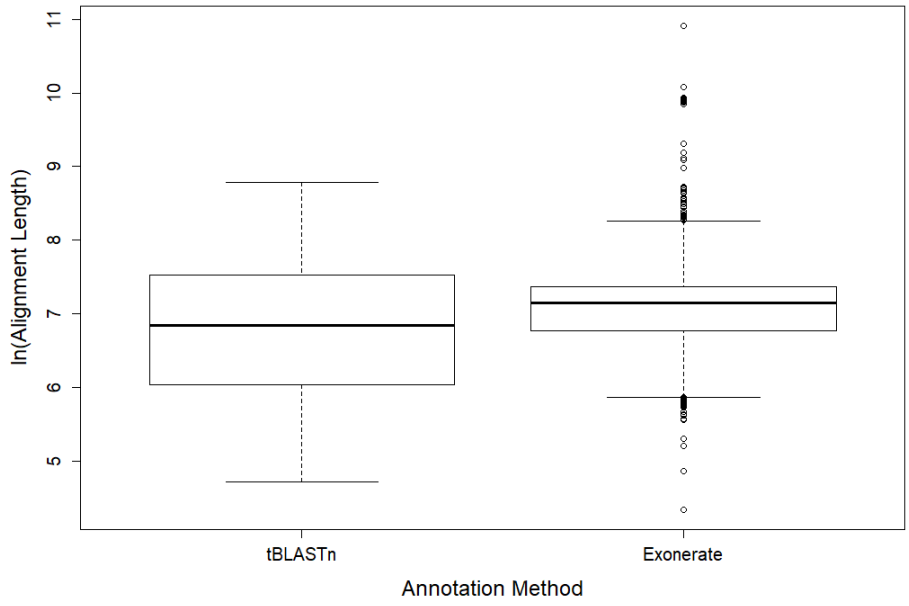

Figure 15. Distribution of alignment lengths. Gene alignment lengths for Exonerate were significantly longer than the alignments produced by tBLASTn for the *H. annuus* proteome and the *L. maackii* genome assembly by the Wilcoxon rank sum test ($p < 0.001$, $W = 1487200000$).

Genome annotation can inform chemical control mechanisms for *L. maackii*

From the BLAST annotation, EPSP synthase (also known as 3-phosphoshikimate 1-carboxyvinyltransferase) was found in the *L. maackii* genome assembly, specifically on scaffold jcf7180006298489 on the positive strand from position 4727 to position 6175. RoundUp, whose active ingredient is glyphosate, inhibits EPSP synthase from properly synthesizing amino acids, ultimately causing plant death. However, overproduction of this enzyme can lead to glyphosate-resistance (Yang et al., 2017).

The gene DAP, which encodes the protein L,L-diaminopimelate aminotransferase, was also found in the *L. maackii* genome assembly, on the positive strand of scaffold jcf1780006347233, starting at position 7733 and ending at position 8095. This enzyme is involved in the biosynthesis of lysine in plants and could be a potential target for an herbicide.

Genome annotation suggests that *L. maackii* may produce rhodoxanthin

Several genes involved in the production of rhodoxanthin from beta-carotene as described by Royer et al. (2020) were found in the *L. maackii* genome assembly by both the BLAST annotation method and the exonerate annotation method. The gene GGPPS (geranylgeranyl pyrophosphate synthase, chloroplastic, accession number A0A251UTT7) was found on scaffold jcf7180006337082 from positions 1794 to 2678 by the exonerate annotation. The BLAST annotation method also found GGPPS on the same scaffold from positions 1797 to 2678. The gene CRTSO (prolycopene isomerase, accession number A0A251UA19) was found on the negative strand of scaffold jcf7180006338224 from positions 93281 to 93448 by the BLAST method but was not found by the exonerate method. Similarly, only the BLAST method detected the gene zds (Zeta-carotene desaturase, accession number Q8H0Q6), specifically on the negative strand of scaffold jcf718000633253 from positions 12306 to 12554. The gene LCYB (putative lycopene

beta cyclase, chloroplastic/chromoplastic, accession number A0A251TZI8) was found by both annotation methods on the positive strand of scaffold jcf7180006315512, differing only in start position. Similarly, both methods detected CCD4-L (Carotenoid cleavage dioxygenase 4-like protein, accession number A0A0K3A5X2) on the negative strand of scaffold jcf7180006317757, differing only in end position. CCD8A and CCD8B (putative carotenoid cleavage dioxygenase 8, accession numbers A0A1Y3BUK2 and A0A251VH23 respectively) were detected by both methods on the negative strand of scaffold jcf7180006337575 in overlapping regions. The final gene involved in the rhodoxanthin production pathway that was identified was CCD7 (putative carotenoid cleavage dioxygenase 7 protein, accession number A0A251SWS7), which found only by the BLAST method on the positive strand of scaffold jcf7180006313603 from position 8804 to 9577. The only genes involved in rhodoxanthin production pathway described by Royer et al. that were not identified by at least one of the annotation methods were phytoene dehydrogenase, beta-carotene hydroxylase-like (BCHL), and beta-carotene hydroxylase (BCH).

Annotation identifies genes that may play a role in plant defense

Three groups of genes of particular interest with respect to plant defense are LOX genes (lipoxygenase), PAL (phenylalanine ammonia lysase), and JAR1 (jasmonoyl-L-amino acid synthetase). Genes from each of these three groups were annotated in the genome assembly for *L. maackii*. Thirty-two hits for PAL genes, some overlapping, were obtained from the BLAST annotation method after filtering, spread across four contigs. For LOX genes, a total of 23 hits were found after filtering from the BLAST annotation method, spread across four contigs. Two hits, overlapping each other, were found for JAR1 by the BLAST annotation method. By the exonerate annotation method, JAR1 was not found, but both PAL and LOX were found. Two separate gene annotations for LOX were found by exonerate, and sixteen gene annotations were

found for PAL, some of which were overlapping. Identification of these genes may aid in the understanding of the defense mechanisms of *L. maackii*.

DISCUSSION

Here I provide a genome assembly for *Lonicera maackii* that is both complete and contiguous for a first pass *de novo* genome assembly. In the contiguity assessment, the disparity between the N50 and the NG50 suggested that the assembly was not complete, but the BUSCO analysis did not support this finding. The BUSCO analysis showed that the genome was in fact complete, having 86.0% of the single copy orthologs conserved among eudicots.

There are a few possible explanations for this difference in completeness findings. Because the NG50 is based on estimated genome size, it is possible that the genome size was overestimated. The genome size was estimated by taking the genome size of a relative with a genome of a known size, *L. japonica* (Chen et al., 2017). Both *L. maackii* and *L. japonica* are known to have diploid chromosome count of 2n=18 (Wang and Wang, 2005), so they were assumed to have about the same genome size. However, if the genome of *L. maackii* is smaller than that of *L. japonica*, that would mean that using the genome size of *L. japonica* was an overestimation of the genome size of *L. maackii*. Overestimating the genome size would cause the NG50 to be much lower than the N50 because the NG50 is calculated based on the estimated genome size and the N50 is calculated based on the assembled genome size.

Another possible explanation for the discrepancy in the N50 and NG50 is that the parts of the genome that are missing in the assembly are not protein coding regions. Plants are known to have extensive regions of repetitive elements, accounting for as much as 90 percent of the genome in some species (Mehrotra and Goyal, 2014). Repetitive elements found in non-coding regions of

the genome would not have been evaluated in the BUSCO analysis because BUSCO evaluates completeness based on gene finding. Furthermore, because of their repetitive nature, repetitive elements are challenging to assemble (Torresen et al., 2019). It is difficult to resolve the difference between overlap and repeated sequences, so it is not unlikely that these regions were not completely and accurately assembled. Taken together, this means that it is possible that the BUSCO analysis could have shown that the genome assembly was complete but that there were in fact many missing non-coding regions, thereby explaining the low NG50 relative to the N50.

One of the major challenges that came with this project was genome annotation. The ideal tool for this step in the project was Exonerate because it provides gene locations along with exons, introns, and splice sites. Additionally, it was found that exonerate produces significantly longer alignments than the tBLASTn alignment output. However, the exonerate program is not parallelized and runs on only one core. The computational challenges associated with this program made it very difficult to annotate the assembled genome within the timeframe of this project. In order to complete the annotation with exonerate within the timeframe of this project, the parameter --seedrepeat had to be used, which reduced the total number of annotations that were generated by exonerate.

The alternative annotation approach involved using tBLASTn. This program completed within two days, but the results were much more difficult to interpret and convert to annotations. Many of the hits that were found were short and corresponded to exons found by Exonerate, but defining what should qualify as an exon versus a duplicated gene was very difficult. Neither method was perfect, and both had their tradeoffs. Additionally, narrowing the BLAST hits by taking the hit(s) with the lowest e-value and disregarding others limits the ability to identify duplicated genes by this method of annotation. The annotations provided here better represent if a

gene is present at all in the genome and does not provide a good measure of the duplication levels of the genes. Future work could be done to work on an annotation method that is able to accurately detect gene duplications.

Another limitation to the annotation methods described here is that annotating based on the proteome of only two species (one in the case of exonerate) limits the genes that can be found. If a protein coding gene is not found in the reference proteome, then it will not be annotated. There are likely many genes that were not annotated for this reason. Additionally, the proteome approach only detects protein-coding genes. Annotating using a large database of genomic sequences could provide a more complete genome annotation.

The gene for EPSP synthase was detected in the *L. maackii* genome assembly by the BLAST annotation method. This gene is the target of glyphosate, the active ingredient in the herbicide RoundUp, which is used in the chemical and hybrid mechanisms for controlling *L. maackii* (Schonbrunn et al., 2001; Fuchs and Geiger, 2005; Gorchov, 2005; Smith and Smith, 2010). Duplications of the EPSPS gene are known to confer glyphosate resistance (Patterson et al., 2018). Because the annotation methods described here do not accurately identify gene duplications, the results from this study do not point to whether or not EPSPS is duplicated in *Lonicera maackii*. However, the fact that this gene was identified suggests that a more focused BLAST search or a more targeted alignment with Exonerate could elucidate whether this individual is resistant to glyphosate. If this were shown, it would suggest that glyphosate is not an adequate herbicide to use for controlling *L. maackii* invasion.

Identifying other potential targets for chemical control and developing alternative herbicides to glyphosate is essential not only because plants are able to develop resistance to glyphosate but also because glyphosate causes toxicological effects in animals (Gill et al., 2017)

L,L-diaminopimelate aminotransferase has been identified as a potential target for herbicides in plants. This enzyme is involved in a pathway for lysine biosynthesis (Hudson et al., 2006). Because this gene was identified in the genome assembly for *L. maackii*, this could be a potential target for a chemical control mechanism alternative to glyphosate. Targeting this protein is also beneficial over targeting EPSPS with glyphosate because humans and plants do not biosynthesize lysine and therefore do not have this pathway (Triassi et al., 2014). Controlling *L. maackii* by targeting L,L-diaminopimelate aminotransferase could help combat the herbicide resistance problem and would likely be safer for animals.

In their recent paper, Royer et al. describe the multistep conversion of beta-carotene to rhodoxanthin in *Lonicera* plants (2020). The enzymes involved in this biosynthesis pathway are geranylgeranyl phosphate synthase, prolycopene isomerase, phytoene dehydrogenase, zeta-carotene desaturase, lycopene cyclase, beta-carotene hydrolase-like, beta-carotene hydrolase, carotenoid cleavage dehydrogenase, and tubulin beta 7,3 chains. Of these enzymes, only three were not identified in the *L. maackii* genome assembly, phytoene dehydrogenase, beta-carotene hydroxylase-like (BCHL), and beta-carotene hydroxylase (BCH). BCHL was not found in the reference for either *H. annuus* or *A. thaliana*, so it is possible that this gene is present but was not found. Additionally, using the --seedrepeat parameter severely limited the annotations that were recorded by exonerate, so it is possible that given more computational resources, these missing genes could be annotated. Understanding the rhodoxanthin biosynthesis pathway in *L. maackii* is one of the first steps toward explaining the aberrant coloring of birds who consume this plant's berries.

Jasmonic acid is a chemical signal used by plants to regulate responses to biotic and abiotic stress. For example, jasmonic acid plays a role in defending the plant from insects and pathogens

and protecting the plant from abiotic stress (Staswick, Tiryaki and Rowe, 2004). Lipoxygenases (LOXs), which initiate fatty acid oxidation, lead to the development of jasmonic acid and other related compounds (Vellosillo et al, 2013). Phenylalanine ammonia-lyase is an inducible enzyme important to a plant's defense against pathogens, UV radiation, and other abiotic and biotic stressors (Kim and Hwang, 2014). Understanding an invasive plant's defense mechanisms can help us to better understand its success in non-native habitat and may even allow us to develop control strategies in light of these defense mechanisms. For example, control strategies that trigger these defense strategies would be less desirable than a control strategy that is known not to trigger a plant's defense.

The first pass *de novo* genome assembly and annotation provided here is the first step toward better understanding *L. maackii* as an invasive species on the genomic level. There are many benefits to studying invasive species using bioinformatics techniques. Genomics can provide a straightforward way to identify species. This is especially important to detecting an invasive species in an area where it has not invaded yet. Intuitively, when identifying the species of an organism, the first species that come to mind are the ones that are known to inhabit that area. Unfortunately, as described here, *Lonicera* resembles many native, harmless shrubs. If a plant was identified in an uninvaded area, it could easily be misidentified based on phenotypic characteristics. However, having a genomic locus or set of loci that can identify the species of an individual provides a fast, unambiguous way to identify the species. If the plant can be accurately identified before a population can become established in a new area, then the threat of the invasive can be controlled and its spread limited.

The bioSAFE (BioSurveillance of Alien Forest Enemies) project has been initiated as an effort to identify invasive species using genomics techniques and to evaluate the risks that these

species may pose (https://www.biosafegenomics.com/). The project has several aims. With genomic information, the team aims to identify invasive species through biosurvelliance efforts, assess the risk that these invasive species pose in the invaded areas, and guide intervention efforts (Bilodeau et al., 2019). Early identification is one of the best ways to control an invasion. For *L. maackii* specifically, there is a short time window when identification can prevent a population from establishing. *L. maackii* plants do not reach reproductive maturation until they are three to eight years old, so if the plants are successfully removed within the first three years, the population will not have had a chance to reproduce (Deering and Vankat, 1999).

The genome assembly and annotation provided here are only a start. This is a first pass, draft assembly. Future work should aim to improve this assembly. Sequencing using a long-read technology could improve the assembly, helping especially with the contiguity (Mantere, Kersten and Hoischen, 2019). Two possible long read technologies are Oxford Nanopore Technologies (ONT) and PacBio. With longer reads, the assembler does not need to connect as many fragments. Murigneux et al. (2020) compared ONT, PacBio, and Illumina short read sequencing. The mean read lengths of ONT, PacBio, and Illumina were 7,962 base pairs, 20,575 base pairs, and 150 (paired end) base pairs respectively. They found that the Illumina assembly had the highest number of missing BUSCO genes. However, the cost to sequence with long read technologies is significantly higher than short read sequencing. In the same study by Murigneux, the cost to sequence on ONT, PacBio, and Illumina was $3,270, $12,560, and $721 respectively.

Annotation could be improved by performing transcriptome assembly instead of genome assembly. The whole genome sequencing performed here gives protein coding regions and non-protein coding regions. However, transcriptome sequencing targets mRNA, so only protein coding genes are found. With whole genome sequencing, the annotation programs look for genes across

the entire genome, but with transcriptome annotation, the program has to look only at the protein coding regions, thereby reducing the computational complexity of the process. Because annotation was performed here using a reference proteome, only protein-coding regions of the genome are needed. The non-protein coding regions of the genome likely contain meaningful information, but given that annotation was only performed for protein-coding genes, having a transcriptome instead of a genome could have improved the annotation.

REFERENCES

Andrews, S. (2019). FastQC: A Quality Control Tool for High Throughput Sequence Data. Retrieved from https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Altschul, S. F., Gish, W., Miller, W., Meyers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403-410.

Ammal, E. K. J. & Saunders, B. (1952). Chromosome numbers in species of Lonicera. *Kew Bulletin*, *7*(4), 539-541.

Bartuszevige, A. M. & Gorchov, D. L. (2006). Avian seed dispersal of an invasive shrub. *Biological Invasions*, *8*(5), 1013-1022.

Bauer, J. T., Shannon, S. M., Stoops, R. E., & Reynolds, H. L. (2012). Context dependency of the allelopathic effects of *Lonicera maackii* on seed germination. *Plant Ecology*, *213*, 1907-1916.

Beck, K. G., Zimmerman, K., Schardt, J. D., & Stone, J. K. (2008). Invasive species defined in a policy context: recommendations from the federal invasive species advisory committee. *Invasive Plant Science and Management*, *1*(4), 414-421.

Bilodeau, P., Roe, A. D., Bilodeau, G., Blackburn, G. S., Cui, M., Cusson, M., Doucet, D., Griess, V. C., Lafond, V. M. A., Nilausen, C., Paradis, G., Porth, I., Prunier, J., Srivastava, V., Steward, D., Torson, A. S., Tremblay, E., Uzunovic, A., Yemshanov, D., & Hamelin, R. C. (2019). Biosurveillance of forest insects: part II – adoption of genomic tools by end user communities and barriers to integration. *Journal of Pest Science*, *92*, 71-82.

CABI, 2020. *Fallopia japonica.* In: Invasive Species Compendium. Wallingford, UK: CAB International. www.cabi.org/isc.

Castellano, S. M. & Gorchov, D. L. (2013). White-tailed Deer (Odocoileus virginianus) disperse seeds of the invasive shrub, amur honeysuckle (Lonicera maackii). *Natural Areas Journal, 33*(1), 778-80.

Chen, J., Xia, N., Wang, X., Beeson, R. C., & Chen, J. (2017). Ploidy level, karyotype, and DNA content in the genus *Lonicera*. *HortScience, 52*(12), 1680-1686.

Cipollini, K., Ames, E., & Cipollini, D. (2009). Amur honeysuckle (Lonicera maackii) management method impacts restoration of understory plants in the presence of white-tailed deer (Odocoileus virginiana). *Invasive Plant Science and Management, 2*(1), 45-54.

Collier, M. H., Vankat, J. L., & Hughes, M. R. (2002). Diminished plant richness and abundance below Lonicera maackii, an invasive shrub. *The American Midland Naturalist, 147*(1), 60-71.

Czarapata, E. J. (2005). *Invasive plants of the upper Midwest: An illustrated guide to their identification and control.* University of Wisconsin Press.

Deering, R. H., & Vankat, J. L. (1999). Forest colonization and developmental growth of the invasive shrub Lonicera maackii. *The American Midland Naturalist, 141*(1), 43-50.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., & Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics, 29*(1), 15-21.

Dolan, B. J. & Parker, G. R. (2004). Understory response to disturbance: an investigation of prescribed burning and understory removal treatments. In M. A. Spetich (Eds.), *Upland oak ecology symposium history, current conditions, and sustainability: Fayetteville, Arkansas, October 7-10, 2002* (285-291). Southern Research Station.

EDDMapS. (2020). Early Detection & Distribution Mapping System. The University of Georgia – Center for Invasive Species and Ecosystem Health. Retrieved from https://www.eddmaps.org/distribution/usstate.cfm?sub=3040

EMBL-EBI. (2020). Exonerate: A generic tool for sequence alignment. Retrieved from https://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate

Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J. F., Dougherty, B. A., & Merrick J. M. (1995). Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science, 269*(5223), 496-512.

Fuchs, M. A. & Geiger, D. R. (2005). Assessing herbicidal damage in amur honeysuckle, Lonicera maackii, stem tissue. In J. Cardina (Ed.), *Ohio invasive plant research conference: Bridging the gap between land management and research* (95-100).

Gardner, A. M., Muturi, E. J., Overmier, L. D., & Allan, B. F. (2017). Large-scale removal of invasive honeysuckle decreases mosquito and avian host abundance. *Ecohealth*, *14*(4), 750-761.

Geiger, D. R., Fuchs, M. A., & Banker, M. G. (2005). Woodland restoration: applied science, natural history, and technology. In J. Cardina (Ed.), *Ohio invasive plant research conference: Bridging the gap between land management and research* (62-70).

Gill, J. P. K., Sethi, N., Mohan, A., Datta, S., Girdhar, M. (2017). Glyphosate toxicity for animals. *Environmental Chemistry Letters*, *16*, 401-426.

Gorchov, D. L. (2005). Control of Invasives and Responses of Native Forest-Floor Plants: case studies of Garlic Mustard and Amur honeysuckle. In J. Cardina (Ed.), *Ohio invasive plant research conference: Bridging the gap between land management and research* (30-42).

Hannon, G. J. (2010). FastX-Toolkit: FASTQ/A short-reads pre-processing tools. Retrieved from http://hannonlab.cshl.edu/fastx_toolkit/

Hartman, K. M. & McCarthy, B. C. (2004). Restoration of a forest understory after the removal of an invasive shrub, amur honeysuckle (*Lonicera maackii*). *Restoration Ecology*, *12*(2), 154-165.

Honeysuckle Spp. (2019, July 2). Retrieved from http://nys.info/invasive_species/honeysuckle

Huang, W. Y., Fu, L., Li, C. Y., Xu, L. P., Zhang, L. X., & Zhang, W. M. (2017). Quercetin, hyperin, and chlorogenic acid improve endothelial function by antioxidant, anti-inflammatory, and ACE inhibitory effects. *Journal of Food Science, 82*(5), 1239-1246.

Hudon, J., Derbyshire, D., Leckie, S., & Flinn, T. (2013). Diet-induced plumage erythrism in Baltimore orioles as a result of the spread of introduced shrubs. *The Wilson Journal of Ornithology*, *125*(1), 88-96.

Hudon, J., Driver, R. J., Rice, N. H., Lloyd-Evans, T. L., Craves, J. A., & Shustack, D. P. (2017). Diet explains red flight feathers in yellow-shafted flickers in eastern North America. *The Auk, 134*(1), 22-33

Hudon, J. & Mulvihill, R. S. (2017). Diet-induced plumage erythrism as a result of the spread of alien shrubs in North America. *North American Bird Bander*, *42*(4), 95-103.

Hudson, A. O., Singh, B. K., Leustek, T., & Gilvarg, C. (2006). An LL-diaminopimelate aminotransferase defines a novel variant of lysine biosynthesis pathway in plants. *Plant Physiology*, *140*(1), 292-301.

Ingold, J. L. and Craycraft, M. J. (1983). Avian frugivory on honeysuckle (*Lonicera*) in Southwestern Ohio in fall. *Ohio Journal of Science, 83*(5), 256-258.

Julia, R., Holland, D. W., & Guenthner, J. (2007). Assessing the economic impact of invasive species: the case of yellow starthistle (Centaurea solsitialis L.) in the rangelands of Idaho, USA. *Journal of Environmental Management, 85*(4), 876-882.

Karsch-Mizrachi, I., Takagi, T., & Cochrane, G. (2018). The international nucleotide sequence database collaboration. *Nucleic Acids Research, 46,* D48-D51.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., & Haussler, D. (2002). The human genome browser at UCSC. *Genome Research, 12*, 996-1006.

Kersey, P. J. (2018). Plant genome sequences: past, present, future. *Current Opinion in Plant Biology, 48,* 1-8.

Kim, D. S. and Hwang, B. K. (2014). An important role of the pepper phenylalanine ammonia-lyase gene (PAL1) in salicylic acid-dependent signalling of the defence response to microbial pathogens. *Journal of Experimental Botany 65*(9), 2295-2306.

Kim, J. W., Lee, Y. S., Seol, D. J., Cho, I. J., Ku, S. K., Choi, J. S., & Lee, H. J. (2018). Anti-obesity and fatty liver-preventing activities of Lonicera caerulea in high-fat diet-fed mice. *International Journal of Molecular Medicine, 42*(6), 3047-3064.

Ko, H., Wei, B., & Chiou, W. (2006). The effects of medicinal plants used in Chinese folk medicine on RANTES secretion by virus-infected human epithelial cells. *Journal of Ethnopharmacology, 107*(2), 205-210.

Kumari, P., Singh, K. P., & Rai, P. K. (2020). Draft genome of multiple resistance donor plant *Sinapis alba*: An insight into SSR, annotations, and phylogenetics. *PLOS ONE, 15*(4), e0231002.

Kuzmin, D. A., Feranchuk, S. I., Sharov, V. V., Cybin, A. N., Makolov, S. V., Putintseva, Y. A., Oreshkova, N. V., & Krutovsky, K. V. (2019). Stepwise large genome assembly approach: a case of Siberian larch (Larix siberica Ledeb). *BMC Bioinformatics, 20,* 37.

Li, H., Hadsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Goncalo, A., Durbin, R., 1000 Genome Project Data Processing Subgroup (2009a). The Sequence Alignment/Map format and SAMtools. *Bioinformatics, 25*(16), 2078-2079.

Lieurance, D. & Cipollini, D. (2012). Damage levels from arthropod herbivores on *Lonicera maackii* suggest enemy release in its introduced range. *Biological Invasions, 14,* 863-873.

Lieurance, D. & Cipollini, D. (2013). Environmental influences on growth and defense responses of the invasive shrub, *Lonicera maackii*, to simulated and real herbivory in the juvenile stage. *Annals of Botony, 112*(4), 741-749.

Lee, C. E. (2002). Evolutionary genetics of invasive species. *Trends in Ecology and Evolution, 17*(8), 386-391.

Lee, S. J., Son, K. H., Chang, H. W., Kang, S. S., & Kim, H. P. (1998). Antiinflammatory activity of Lonicera japonica. *Phytotherapy Research, 12*(6), 445-447.

Lowe, S., Browne, M., Boudjelas, S., & De Poorter, M. (2000). *100 of the world's worst invasive alien species: A selection from the global invasive species database.* Invasive species specialist group.

Luken, J. O., & Goessling, N. (1994). Seedling distribution and potential persistence of the exotic shrub Lonicera maackii in fragmented forests. *American Midland Naturalist, 133*(1), 124-130.

Luken, J. O. & Thieret, J. W. (1995). Amur honeysuckle (Lonicera maackii; Caprifoliaceae): its ascent, decline, and fall. *Sida, 16*(3), 479-503.

Luken, J. O. & Thieret, J. W. (1996). Amur honeysuckle, its fall from grace. *Bioscience, 46*(1), 18-24.

Mantere, T., Kersten, S., & Hoischen, A. (2019). Long-read sequencing emerging in medical genetics. *Frontiers in Genetics, 10*, 426.

Maron, J. L. & Vila, M. (2001). When do herbivores affect plant invasion? Evidence for the natural enemies and biotic resistance hypotheses. *Oikos, 95*(3), 361-373.

McEwan, R. W., Birchfield, M. K., Schoergendorfer, A., & Arthur M. A. (2009). Leaf phenology and freeze tolerance of the invasive shrub Amur honeysuckle and potential native competitors. *Journal of the Torrey Botanical Society, 136*(2), 212-220.

McKenna, D., Scully, E. D., Pauchet, Y., Hoover, K., Kirsch, R., Geib, S. M., Mitchel, R. F., Waterhouse, R. M., Ahn, S., Arsala, D., Benoit, J. B., Blackmon, H., Bledsoe, T., Bowsher, J. H., Busch, A., Calla, B., Chao, H., Childers, A. K., Childers, C., …, Richards, S. (2016). Genome of the Asian longhorned beetle (Anaplophora galbripennis), a globally significant invasive species, reveals key functional evolutionary innovations at the beetle-plant interface. *Genome Biology, 17*(1), 227-245.

McNeish, R. E. and McEwan, R. W. (2016). A review on the invasion ecology of Amur honeysuckle (*L. maackii*, Caprifoliaceae) a case study of ecological impacts at multiple scales. *Journal of the Torrey Botanical Society, 143*(4), 367-385.

Mehrotra, S. and Goyal, V. (2014). Repetitive sequences in plant nuclear DNA: Types, distribution, evolution and function. *Genomics Proteomics Bioinformatics, 12*(4), 164-171.

Miller, K. E. and Gorchov, D. L. (2004). The invasive shrub, Lonicera maackii, reduces growth and fecundity of perennial forest herbs. *Oecologia, 139*(3), 359-375.

Mochida, K., Sakurai, T., Seki, H., Yoshida, T., Takahagi, K., Sawai, S., Uchiyama, H., Muranaka, T., & Saito, K. (2016). Draft genome assembly and annotation of *Glycyrrhiza uralensis*, a medicinal legume. *The Plant Journal, 89*(2), 181-194.

Moerman, D. E. (1989). Poisoned apples and honeysuckles: the medicinal plants of Native America. *Medical Anthropology Quarterly, 3*(1), 52-61.

Murigneux, V., Rai, S. K., Furtado, A., Bruxner, T. J. C., Tian, W., Harliwong, I., Wei, H., Yang, B., Ye, Q., Anderson, E., Mao, Q., Drmanac, R., Wang, O., Peters, B. A., Xu, M., Wu, P., Topp, B., Coin, L. J. M., & Henry, R. J. (2020). Comparison of long-read methods for sequencing and assembly of a plant genome. *GigaScience, 9*(12), giaa146.

Nakamura, T., Nakazawa, Y., Onizuka, S., Satoh, S., Chiba, A., Sekihashi, K., Miura, A., Yasugahira, N., & Sasaki, Y. F. (1997). Antimutagenicity of tochu tea (an aqueous extract of Eucommia ulmoides leaves): 1. The clastogen-suppressing effects of tochu tea in CHO cells and mice. *Mutation Research, 388*(1), 7-20.

Nie, X., Lv, S., Zhang, Y., Du, X., Wang, L., Biradar, S. S., Tan, X., Wan, F., & Weining, S. (2012). Complete chloroplast genome sequence of a major invasive species, crofton weed (Ageratina adenophora). *PLoS One, 7*(5), e36869.

Nikzad-Langerodi, R., Ortmann, S., Pferschy-Wenzig, E. M., Bochkov, V., Zhao, Y. M., Miao, J. H., Saukel, J., Ladurner, A., Heiss, E. H., Dirsch, V. M., Bauer, R., & Atanasov, A. G. (2017). Assessment of anti-inflammatory properties of extracts from Honeysuckle (Lonicera sp. L., Caprifoliaceae) by ATR-FTIR spectroscopy. *Talanta, 175,* 264-272.

Pandit, M. K., Pocock, M. J. O., & Kunin, W. E. (2011). Ploidy influences rarity and invasiveness in plants. *Journal of Ecology, 99*(5), 1108-1115.

Patterson, E. L., Pettinga, D. J., Ravet, K., Neve, P., & Gaines, T. A. (2018). Glyphosate resistance and EPSPS gene duplication: convergent evolution in multiple plant species. *Journal of Heredity, 109*(2), 117-125.

Peebles-Spencer, J. R., Gorchov, D. L., & Crist, T. O. (2017). Effects of an invasive shrub, Lonicera maackii, and a generalist herbivore, white-tailed deer, on forest floor plant community composition. *Forest Ecology and Management, 402,* 204-212.

Peng, Y., Lai Z., Lane, T., Nageswara-Rao, M., Okada, M., Jasieniuk, M., O'Geen, H., Kim, R. W., Sammons, D., Rieseberg, L. H., & Stewart, C. N. (2014). De novo genome assembly of the evonomically important weed horseweed using integrated data from multiple sequencing platforms. *Plant Physiology, 166,* 1241-1254.

Prior, K. M., Powell, T. H., Joseph, A. L., & Hellmann, J. J. (2015). Insights from community ecology into the role of enemy release in causing invasion success: the importance of native enemy effects. *Biological invasions, 17*(5), 1283-1297.

Pysek, P., Skalova, H., Cuda, J., Guo, W., Suda, J., Dolezal, J., Kauzal, O., Lambertini, C., Lucanova, M., Mandakova, T., Moravcova, L., Pyskova, K., Brix, H., & Meyerson, L. A. (2018). Small genome separates native and invasive populations in an ecologically important cosmopolitan grass. *Ecology, 99*(1), 79-90.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Comptuing, Vienna, Austria. https://www.R-project.org/.

Royer, J., Shanklin, J., Balch-Kenney, N., Mayorga, M., Huston, P., de Jong, R. M., McMahon, J., Laprade, L., Blomquist, P., Berry, T., Cai, Yuanheng, LoBuglio, K., Trueheart, J., & Chevreux, B. (2020). Rhodoxanthin synthase from honeysuckle; a membrane diiron enzyme catalyzes the multistep conversion of beta-carotene to rhodoxanthin. *Science Advances, 6*(17), eaay9226.

RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA. http://www.rstudio.com/

Sasaki, Y. F., Chiba, A., Murakami, M., Sekihashi, K., Tanaka, M., Takahoko, M., Moribayashi, S., Kudou, C., Hara, Y., Nakazawa, Y., Nakamura, T., & Onizuka, S. (1996). Antimutagenicity of tochu tea (an aqueous extract of Eucommia ulmoides leaves): 2. Suppressing effect of tochu tea on the urine mutagenicity after ingestion of raw fish and cooked beef. *Mutation Research, 371*(3-4), 203-214.

Schonbrunn, E. Eschenburg, S., Shuttleworth, W. A., Schloss, J. V., Amrhein, N., Evans, J. N. S., & Kabsch, W. (2001). Interaction of the herbicide glyphosate with its target enzyme 5-enolpyruvylshikimate 3-phosphate synthase in atomic detail. *PNAS, 98*(4), 1376-1380.

Shi, H., Li, W., & Xu, Xin. (2016). Learning the Comparning and Converting Method of Sequence Phred Quality Score. In *Proceedings of the 2016 6th International Conference on Management, Education, Information and Control (MEICI 2016)*(260-263). Atlantic Press.

Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., & Zdobnov, E. M. (2015). BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics, 31*(19), 3210-3212.

Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. M., & Birol, I. (2009). ABySS: A parallel assembler for short read sequence data. *Genome Research, 19*, 1117-1123.

Slater, G. S. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics, 6*(31).

Smith, K. & Smith, A. (2010). Controlling non-native invasive plants in Ohio forests: Bush honeysuckles. *Agriculture and Natural Resources*, F68.

Smith, S. B., DeSando, S. A., & Pagano, T. (2013). The value of native and invasive fruit-bearing shrubs for migrating songbirds. *Northeastern Naturalist, 20*(1), 171-184.

Staswick, P. E., Tiryaki, I., & Rowe, M. L. (2002). Jasmonate response locus JAR1 and several related Arabidopsis genes encode enzymes of the firefly luciferase superfamily that show activity on jasmonic, salicylic, and indole-3-acetic acids in an assay for adenylation. *Plant Cell, 14*(6), 1405-1415.

The Arabidopsis Genome Initiative. (2000). Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature, 408*, 796-815.

Torresen, O. K., Star, B., Mier, P., Andrade-Navarro, M. A., Bateman, A., Jarnot, P., Gruca, A., Grynberg, M., Kajava, A. V., Prompona, V. J., Anisimova, M., Jakobsen, K. S., & Linke, D. (2019). Tandem repeats lead to sequence assembly errors and impose multi-level challenges for genome and protein databases. *Nucleic Acid Research, 47*(21), 10994–11006.

Triassi, A. J., Wheatley, M. S., Savka, M. A., Gan, H. M., Dobson, R. C. J., & Hudson, A. O. (2014). L,Ll-diaminopimelate aminotransferase (DapL): a putative target for the development of narrow-spectrum antibacterial compounds. *Frontiers in Microbiology, 5*, 509.

Trisel, D. E. (1997). The invasive shrub, Lonicera maackii (Rupr.) Herder (Caprifoliaceae): Factors contributing to its success and its effect on native species. Ph.D. Dissertation, Miami Univeristy, Oxford, Ohio.

Trisel, D. E. & Gorchov, D. L. (1994). Regional distribution, leaf phenology, and herbivory of the invasive shrub, Lonicera maackii. *Bulletin of the Ecological Society of America, 75*, 231-232.

Twyford, A. D. (2018). The road to 10,000 plant genomes. *Nature Plants, 4*(6), 312-313.

Vellosillo, T., Aguilera, V., Marcos, R., Bartsch, M., Vicente, J., Cascon, T., Hamberg, M., & Castresana, C. (2013). Defense activated by 9-lipoxygenase-derived oxylipins requires specific mitochondrial proteins. *Plant Physiology 161*, 617-627.

Wang, F. & Wang, B. (2005). Karyotype Analysis of Lonicera Japonica and L. maackii. *Zhong Yao Cai, 28*(3), 168-170.

Ward, S. M., Gaskin, J. F., & Wilson, L. M. (2008). Ecological genetics of plant invasions: What do we know? *Invasive Plant Science and Management, 1*(1), 98-109.

Watling, J. I., Hickman, C. R., Orrock, J. L. (2011). Invasive shrub alters native forest amphibian communities. *Biological Conservation, 144*, 2597-2601.

Western New York Partnership for Regional Invasive Species Management (WNY PRISM) (2021). Autumn Olive. *Western New York PRISM*. https://www.wnyprism.org/invasive_species/autumn-

olive/#:~:text=Autumn%20olive%20was%20introduced%20to,%2Fanimals%2F99141.html.

Yang, X., Beres, Z. T., Jin, L., Parrish, J. T., Zhao, W., Mackey, D., Snow, A. A. (2017). Effects of over-expressing a native gene encoding 5-enolpyruvylshikimate-3-phosphate synthase (EPSPS) on glyphosate resistance in Arabidopsis thaliana. *PLOS ONE, 12*(4), e0175820. s

Zhang, Y., Li, L., Yan, T. L., & Liu, Q. (2014) Complete chloroplast genome sequences of Praxelis (Eupatorium catarium Veldkamp), an important invasive species. *Gene, 549*(1), 58-69.

Zhang, Y., Zheng, L., Zheng, Y., Zhou, C., Huang, P., Xiao, X., Zhao, Y., Hao, Z., Hu, Z., Chen, Q., Li, H., Wang, X., Fukushima, K., Wang, G., & Li, C. (2019). Assembly and annotation of a draft genome of the medicinal plant *Polygonum cuspidatum*. *Frontiers in Plant Science, 10*.

Zhang, T., Qiao, Q., Novikova, P. Y., Wang, Q., Yue, J., Guan, Y. Ming, S., Liu, T., De, J., Liu, Y., Al-Shehbaz, I. A., Sun, H., Van Montagu, M., Huang, J., Van de Peer, Y., & Qiong, L. (2019). Genome of Crucihimalaya himalaica, a close relative of Arabidopsis, shows ecological adaptation to high altitude. *Proceedings of the National Academy of Science, 116*(14), 7137-7146.

Zhang, B., Yang, R., Zhao, Y., & Liu, C. Z. (2008). Separation of chlorogenic acid from honeysuckle crude extracts by macroporous resins. *Journal of Chromatography B Analytic Technologies in the Biomedical and Life Sciences, 867*(2), 253-258.

Zhou, Z., Li, X., Liu, J., Dong, L., Chen, Q., Liu, J., Kong, H., Zhang, Q., Qi, X., Hou, D., Zhang, L., Zhang, G., Liu, Y., Zhang, Y., Li, J., Wang, J, Chen, X., Wang, H., Zhang, J., …, Zhang, C. Y. (2015). Honeysuckle-encoded atypical microRNA2911 directly targets influenza A viruses. *Cell Research, 25*(1), 39-49.

Zimin, A. V., Marcais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics, 29*(21), 2669-2677.

Zouhar, K., Smith, J. K. & Sutherland, S. (2008). Chapter 2: Effects of fire on nonnative invasive plants and invasibility of wildland ecoystems. In Zouhar, K., Smith, J. K., Sutherland, S., Brooks, M. L., *Wildland fire in ecosystems: fire and nonnative invasive plants* (7-32). U. S. Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Appendix A

Scripts and Programs

SCRIPT / PROGRAM NAME:

runFASTQC.sh

PURPOSE:

Run the quality assessment program FASTQC for raw fastq files.

APPLICATION IN THIS PROJECT:

The FASTQC program produces detailed output for each of the fastq input files, describing the overall quality of the data. This program was run on the raw sequencing data before any pre-processing to determine the initial quality of the data. While some assemblers require that the information from these quality reports be used to pre-process the raw sequences, no pre-processing was done for this project aside from removing contamination from chloroplast and mitochondrial reads because MaSuRCA does not require pre-processing.

USAGE INFORMATION:

Execute the bash script from the directory containing the raw sequencing reads: `./runFASTQC.sh`

```
######################### runFASTQC.sh #########################

#!/bin/bash

# FASTQC for raw sequencing reads

# create directory for fastqc output
mkdir fastqc_out
cd fastqc_out

# run fastqc on all of the raw fastq files
/usr/local/bin/FASTQC_11.9/fastqc ../*.fq.gz
```

```
######################### runFASTQC.sh #########################
```

SCRIPT / PROGRAM NAME:

chloroAlign.sh

PURPOSE:

Use STAR to align the reads to the chloroplast genome of *Lonicera maackii* and write the unmapped reads to new fastq files.

APPLICATION IN THIS PROJECT:

Even though the isolation of the nuclear DNA was not meant to include any plastid DNA, the raw reads were aligned to the chloroplast genome to remove any reads generated from contamination of the genomic DNA with chloroplast DNA. The program runs STAR sequentially on each of the input FASTQ files and generates a new FASTQ file with only reads that did not map to the chloroplast genome.

USAGE INFORMATION:

The input to this program is the raw FASTQ files, and the output of this program is FASTQ files with chloroplast reads removed.

Before running this script, create a sub-directory for the output. To run this script, execute the script from the parent directory containing the output sub-directory and the chloroplast FASTA file:

```
./chloroAlign.sh
```

```
######################## chloroAlign.sh ##########################

#!/bin/bash

# STAR alignment for chloroplast reference genome

echo "running script $0"
echo "starting time is `date`"

# Step 1: generate genome index files

echo "generating genome index files..."
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir
genomeDir -genomeFastaFiles lonicera_maackii_chloroplast.fasta

# Step 2: mapping reads to the genome

cd STAR_output

for i in ../../rawReads/*_1.fq; do
      echo "starting alignment of file pair ${i:15:35}..."
      date
      /usr/local/bin/STAR/STAR/ --runThreadN 8 --runMode alignReads
      --outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir
      ../genomeDir --readFilesIn $i ${i%_1.fq}_2.fq --outFileNamePrefix
      ${i:15:35} --outReadsUnmapped Fastx
      echo "finished alignment of file pair ${i:15:35}..."
      date
done

echo "ending time is `date`"

######################## chloroAlign.sh ##########################
```

SCRIPT / PROGRAM NAME:

mitoAlign.sh

PURPOSE:

Use STAR to align the reads that did not map to the chloroplast genome of *Lonicera maackii* to the mitochondrial genome of *H. annuus* and write out unmapped reads to new FASTQ files.

APPLICATION IN THIS PROJECT:

It was possible that isolated nuclear DNA was contaminated with some mitochondrial DNA, so this script was written to use STAR to remove these reads. Prior to running this script, chloroplast reads were removed, and unmapped reads were written to new FASTQ files, which were then used as input for this script. No mitochondrial genome was available for *L. maackii*, so the most closely related mitochondrial genome available at the time this script was run, that of *H. annuus*, was used.

USAGE INFORMATION:

The input to this program is the FASTQ files with chloroplast reads removed, and the output of this program is FASTQ files containing reads only for nuclear DNA (chloroplast and mitochondrial DNA reads removed). Before running this script, create a sub-directory for the STAR output. To run this script, execute it within the parent directory containing the output sub-directory and the *H. annuus* mitochondrial genome FASTA file:

```
./mitoAlign.sh
```

```
########################## mitoAlign.sh ##########################

#!/bin/bash

# STAR alignment for mitochondrial reference genome

echo "running script $0"
echo "starting time is `date`"

# step 1: generate genome index files

echo "generating genome index files..."
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir
genomeDir --genomeFastaFiles Helianthus_annuus_mito.fasta

# Step 2: mapping reads to the genome

cd STAR_output

for i in ../../chloro_align/STAR_output/*Unmapped.out.mate1; do
      ehco "starting alignment of file pair ${i:31:35}..."
      date
      /usr/local/bin/STAR/STAR --runThreadN 8 --runMode alignReads
      --outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir
      ../genomeDir --readFilesIn $i ${i%1}2 --outFileNamePrefix ${i:31:35}
      --outReadsUnmapped Fastx
      echo "finished alignment of file pair ${i:31:35}..."
      date
done

echo "Ending time is `date`"

########################## mitoAlign.sh ##########################
```

SCRIPT / PROGRAM NAME:

find_species.sh

PURPOSE:

Use STAR to align the raw reads of two FASTQ files to the chloroplast genomes of *L. maackii* and *Elaeagnus macrophylla*.

APPLICATION IN THIS PROJECT:

Chloroplast alignment indicated that the nuclear DNA was contaminated with chloroplast DNA. Therefore, the chloroplast reads can be used to support the species identification. Autumn olive, an invasive species found also found in Western New York, has a somewhat similar appearance to the bush honeysuckles. Reads were aligned to the chloroplast genome of *E. macrophylla*, a relative of autumn olive and to that of *L. maackii* to determine the chloroplast of which species the sequencing data more closely represents.

USAGE INFORMATION:

The input of this script is two raw FASTQ files. Important output generated by STAR is found in the {prefix}Log.final.out files, which contain a summary of the alignment. Before running this script, create a sub-directory for the STAR output for each species. The parent directory should contain the two output sub-directories, the chloroplast genome FASTA files for *L. maackii* and *E. macryphylla*, and the FASTQ files to be used. To run this script, execute it within the parent directory:

```
./find_species.sh
```

```
######################### find_species.sh ########################

#!/bin/bash

# STAR alignment for 2 chloroplast references

# alignmet 1: L. maackii

echo "running STAR for L. maackii"
date

# Step 1: generate genome index files
echo "generating genome index files..."
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir
genomeDirL --genomeFastaFiles lonicera_maackii_chloroplast.fasta

# step 2: mapping reads to the genome
cd L_m_STAR_out

echo "starting alignment of file pair one..."
date
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode alignReads --
outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir ../genomeDirL -
-readFilesIn ../V300022786_L2_B5GHONrknDAAAAAAA-501_1.fq
../V300022786_L2_B5GHONrknDAAAAAAA-501_2.fq --outFileNamePrefix
V300022786_L2_B5GHONrknDAAAAAAA-501 --outReadsUnmapped Fastx
echo "finished alignment of file pair one..."
date

echo "starting alignment of file pair two..."
date
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode alignReads --
outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir ../genomeDirL -
-readFilesIn ../V300022786_L2_B5GHONrknDAAAAAAA-502_1.fq
../V300022786_L2_B5GHONrknDAAAAAAA-502_2.fq --outFileNamePrefix
V300022786_L2_B5GHONrknDAAAAAAA-502 --outReadsUnmapped Fastx
echo "finished alignment of file pair two..."
date

######

# alignment for autumn olive relative
echo "running STAR for E. macrophylla"
date

cd ..

# Step 1: generate genome index files
echo "generating genome index files..."
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir
genomeDirE --genomeFastaFiles Elaeagnus_macrophylla_chloro.fasta

# step 2: mapping reads to the genome
cd E_m_STAR_out

######################### find_species.sh ########################
```

```
######################### find_species.sh ########################

echo "starting alingment of file pair one..."
date
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode alignReads --
outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir ../genomeDirE -
-readFilesIn ../V300022786_L2_B5GHONrknDAAAAAAA-501_1.fq
../V300022786_L2_B5GHONrknDAAAAAAA-501_2.fq --outFileNamePrefix
V300022786_L2_B5GHONrknDAAAAAAA-501 --outReadsUnmapped Fastx
echo "finished alignment of file pair one..."

echo "starting alignment of file pair two..."
date
/usr/local/bin/STAR/STAR --runThreadN 8 --runMode alignReads --
outFilterMismatchNmax 2 --outSAMtype BAM Unsorted --genomeDir ../genomeDirE -
-readFilesIn ../V300022786_L2_B5GHONrknDAAAAAAA-502_1.fq
../V300022786_L2_B5GHONrknDAAAAAAA-502_2.fq --outFileNamePrefix
V300022786_L2_B5GHONrknDAAAAAAA-502 --outReadsUnmapped Fastx
echo "finished alignment of file pair two..."
date
```

######################### find_species.sh ########################

SCRIPT / PROGRAM NAME:

speciesBLAST.sh

PURPOSE:

Use a nucleotide BLAST to find regions in the chloroplast reads where four *trn* genes align.

APPLICATION IN THIS PROJECT:

Plants classified as Elaeagnaceae, including *E. macrophylla*, have been shown to have a gene duplication of the *trnH* gene in the chloroplast genome. There is not evidence of duplication of this gene in the chloroplast genome of *L. maackii*. To further corroborate the species identification as *L. maackii*, three other *trn* genes known to be not duplicated in either species were identified. A BLAST search was performed for the four *trn* genes against the raw reads to try to quantify the duplication levels of these four trnH genes.

USAGE INFORMATION:

Run this script from the parent directory, containing the STAR output directories for the alignment to the *L. maackii* chloroplast and for the alignment to the *E. macrophylla* chloroplast:

```
./speciesBLAST.sh
```

```
######################### speciesBLAST.sh ########################

#!/bin/bash

# run for L_m output
cd L_m_STAR_out

# convert bam file to fasta
/usr/local/bin/samtools fasta V300022786_L2_B5GHONrknDAAAAAAA-
502Aligned.out.bam > V300022786_L2_B5GHONrknDAAAAAAA-502_aligned_test.fa

#Make  BLAST database from fragments
/usr/local/bin/blastplus/makeblastdb -in V300022786_L2_B5GHONrknDAAAAAAA-
502_aligned_test.fa -input_type fasta -dbtype nucl -out chloro_blast_db -
title "Chloroplast Nt Blast DB"

#Run nucleotide blast using the genes as the query sequence
/usr/local/bin/blastplus/blastn -db chloro_blast_db -query ../trn_genes.fa -
out chloro_blast.out -num_threads 6 -outfmt "6 qseqid sseqid pident nident
length slen qlen mismatch gaps evalue bitscore" -max_target_seqs 5000


########

# run for E_m output
cd ../E_m_STAR_out

# convert bam file to fasta
/usr/local/bin/samtools fasta V300022786_L2_B5GHONrknDAAAAAAA-
502Aligned.out.bam > V300022786_L2_B5GHONrknDAAAAAAA-502_aligned_test.fa

# Make blast database from fragments
/usr/local/bin/blastplus/makeblastdb -in V300022786_L2_B5GHONrknDAAAAAAA-502-
aligned_test.fa -input_type fasta -dbtype nucl -out chloro_blast_db -title
"Chloroplast Nt Blast DB"

#Run nucleotide blast using the genes as the query sequence
/usr/local/bin/blastplus/blastn -db chloro_blast_db -query ../trn_genes.fa -
out chloro_blast_Em.out -num_threads 6 -outfmt "6 qseqid sseqid pident nident
length slen qlen mismatch gaps evalue bitscore" -max_target_seqs 5000

######################### speciesBLAST.sh ########################
```

SCRIPT / PROGRAM NAME:

trnAnalysis.R

PURPOSE:

Estimate the relative gene duplication level of four trn genes in the aligned chloroplast reads.

APPLICATION IN THIS PROJECT:

The *trnH* gene has been shown to be duplicated in the chloroplast of *E. macropyhlla* but not in *L. maackii*. Quantifying the duplication level as the relative number of hits for *trnH* and three other *trn* genes known not to be duplicated in either species was used as further evidence to support the DEC officer's species identification.

USAGE INFORMATION:

The input for this script is the output from the speciesBLAST.sh script. Execute the commands from this R script within RStudio.

```
######################### trnAnalysis.R ##########################

# script to analyze the output from blast search for molecular
# support of species based on chloroplast genes

# read in the blast output
dat = read.table("chloro_blast.out", header = F, stringsAsFactors = F, sep =
"\t",
                col.names = c("qseqid", "sseqid", "pident", "nident", "length",
                              "slen", "qlen", "mismatch", "gaps", "evalue",
"bitscore"))
# keep only hits that have a mismatch of 0
dat2 = dat[dat$mismatch == 0,]

# keep hits with no gaps
dat3 = dat2[dat2$gaps == 0,]

# keep hits with alignment length at least 80% of the query sequence length
dat4 = dat3[dat3$length/dat3$qlen >= 0.8,]

# plot the duplication levels
levels = as.data.frame(table(dat4$qseqid), stringsAsFactors = F)
levels.plot = levels[grep("Lm", levels$Var1),]
barplot(levels.plot$Freq, xlab = "Gene", ylab = "Blastn Hits", ylim = c(0,2000),
cex.lab = 1.3, names = c("trnH-GUG", "trmM-CAU", "trnQ-UUG", "trnW-CCA"))
abline(h = 0)

######################### trnAnalysis.R ##########################
```

SCRIPT / PROGRAM NAME:

sr_config.txt

PURPOSE:

This is the configuration file for MaSuRCA genome assembler and contains the user-defined parameters to use during assembly.

APPLICATION IN THIS PROJECT:

The parameters used for *de novo* genome assembly with MaSuRCA were entered into the configuration file. The configuration file contains two sections, data and parameters. In the data section, the path to the FASTQ files is provided. The input FASTQ files used for this file are the files containing the reads that did not map to either the chloroplast or the mitochondrial reference sequences. In the parameters section, the assembly parameters are given.

USAGE INFORMATION:

The configuration file is supplied to the masurca program, which generates the assembly script. To generate the assembly script, run the following command from the directory containing the configuration file:

```
/usr/local/bin/MaSuRCA/bin/masurca sr_config.txt
```

This command generates a shell script called assemble.sh in the directory where the command was run. To run the assembly, execute the new script:

```
./assemble.sh
```

```
########################## sr_config.txt ##########################

DATA
# Entered in the format: PE= prefix length sd mate1.fq mate2.fq
PE= ab 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate2
PE= ac 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate2
PE= ad 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate2
PE= ae 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate2
PE= af 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate2
PE= ag 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate2
PE= ah 200 1 ../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate1
../mito_align/STAR_output/V300022786_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate2
PE= ai 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate2
PE= aj 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate2
PE= ak 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate2
PE= al 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate2
PE= am 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate2


########################## sr_config.txt ##########################
```

```
########################## sr_config.txt ##########################
PE= an 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate2
PE= ao 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate2
PE= ap 200 1 ../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate1
../mito_align/STAR_output/V300031453_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate2
PE= aq 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate2
PE= ar 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate2
PE= as 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate2
PE= at 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate2
PE= au 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate2
PE= av 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate2
PE= aw 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate2
PE= ax 200 1 ../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate1
../mito_align/STAR_output/V300032931_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate2
PE= ay 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate2
PE= az 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate2

########################## sr_config.txt ##########################
```

```
########################## sr_config.txt ##########################

PE= ba 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate2
PE= bb 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate2
PE= bc 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate2
PE= bd 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate2
PE= be 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate2
PE= bf 200 1 ../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate1
../mito_align/STAR_output/V300032947_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate2
PE= bg 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
501Unmapped.out.mate2
PE= bh 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
502Unmapped.out.mate2
PE= bi 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
503Unmapped.out.mate2
PE= bj 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
504Unmapped.out.mate2
PE= bk 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
505Unmapped.out.mate2
PE= bl 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
506Unmapped.out.mate2
PE= bm 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
507Unmapped.out.mate2

########################## sr_config.txt ##########################
```

```
######################### sr_config.txt ########################

PE= bn 200 1 ../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate1
../mito_align/STAR_output/V300033085_L2_B5GHONrknDAAAAAAA-
508Unmapped.out.mate2
END

PARAMETERS
EXTEND_JUMP_READS=0
GRAPH_KMER_SIZE = auto
USE_LINKING_MATES = 1
LIMIT_JUMP_COVERAGE = 300
CA_PARAMETERS = cgwErrorRate=0.15
CLOSE_GAPS=1
NUM_THREADS = 8
JF_SIZE = 60000000000
SOAP_ASSEMBLY=0
END

######################### sr_config.txt ########################
```

SCRIPT / PROGRAM NAME:

get_contig_stats.sh

PURPOSE:

Use the AbySS program abyss-fac to retrieve contiguity statistics from the final assembly output FASTA files.

APPLICATION IN THIS PROJECT:

Contiguity statistics are used to evaluate how continuous a genome assembly is, mainly focusing on the number of contigs and their lengths. MaSuRCA generates two FASTA files, one containing contigs, which are assembled reads, and one containing scaffolds, which are assembled contigs. This script runs the program abyss-fac to retrieve the contiguity statistics for the contigs, genome.ctg.fasta, and the scaffolds, genome.scf.fasta.

USAGE INFORMATION:

The input files to this script are the two FASTA files created by MaSuRCA, genome.ctg.fasta and genome.scf.fasta. A tabular file with the statistics for each input file is generated as output. Before running this script, create a contiguity directory within the assembly parent directory and place this script in the new directory.

Execute this script from within the newly created contiguity directory:

```
./get_contig_stats.sh
```

```
###################### get_contig_stats.sh ####################

#!/bin/bash

# use abyss-fac to get stats
/usr/local/bin/ABySS_nonmpi/bin/abyss-fac -G 2270000000 -v -t 1 ../CA/10-
gapclose/genome.ctg.fasta ../CA/10-gapclose/genome.scf.fasta >
abyss_fac_stats.out

###################### get_contig_stats.sh ####################
```

SCRIPT / PROGRAM NAME:

runBUSCO.sh

PURPOSE:

Use BUSCO to evaluate the completeness of the assembled genome.

APPLICATION IN THIS PROJECT:

Because this is a *de novo* genome assembly and little is known about the genome of the species, there is no way to know how much of the genome has been assembled. BUSCO uses genes that are expected to be constant among a group of organisms to evaluate how complete an assembly is. A very complete assembly will have many of the BUSCO genes for the relevant lineage. In a less complete assembly, more of the BUSCO genes would be missing or fragmented. The lineage that best matches *L. maackii* is the *eudicots_odb10* lineage.

USAGE INFORMATION:

The input file for this script is the final assembly output, genome.scf.fasta. The output of the program tells what percentage of BUSCOs were found in their complete form (single copy or duplicated), the percentage that were fragmented, and the percentage that were missing. Create a new directory for completeness within the parent assembly directory and execute this script from within that directory:

```
./runBUSCO.sh
```

```
########################## runBUSCO.sh ############################

#!/bin/bash

# export the path to the config file for busco
export
BUSCO_CONFIG_FILE=~/Honeysuckle_genome/F19FTSUSAT1275_HONrknD/Clean/PL2-
1/busco-4.1.4/config/config.ini

# export paths for augustus
export PATH=/usr/bin/augustus:$PATH
export PATH=/usr/share/augustus/scripts:$PATH
export
AUGUSTUS_CONFIG_PATH=~/Honeysuckle_genome/F19FTSUSAT1275_HONrknD/Clean/PL2-
1/busco-4.1.4/augustus_config_dir/

# run BUSCO using the eudicots_odb10 lineage
~/Honeysuckle_genome/F19FTSUSAT1275_HONrknD/Clean/PL2-1/busco-4.1.4/bin/busco
-i ../CA/10-gapclose/genome.scf.fasta -l eudicots_odb10 -o
BUSCO_eudicot_out.out -m genome -f

########################## runBUSCO.sh ############################
```

SCRIPT / PROGRAM NAME:

runExonerate.sh

PURPOSE:

Use Exonerate to annotate the assembled genome based on the proteome for *H. annuus*. Exonerate provides gene locations as well as coding sequences, exons, introns, and splice sites.

APPLICATION IN THIS PROJECT:

The annotations are meant to give meaning to the genome assembly. After the genome was assembled and that assembly was evaluated for completeness and contiguity, this script was used to identify probable gene locations. The proteome of *H. annuus* was used as a basis for annotation because, at the time of this analysis, this was the most closely related reference species available.

USAGE INFORMATION:

The input for this script is the reference proteome for *H. annuus* and the genome.scf.fasta output file from MaSuRCA to create the annotations. The output from this script is a GFF annotation file. The output from exonerate is mostly in GFF format, but an intermediate command is used to convert the exonerate output to a GFF file.

Execute this script from a directory containing the *H. annuus* proteome and the genome.scf.fasta file:

```
./runExonerate.sh
```

```
####################### runExonerate.sh ########################

#!/bin/bash

# run exonerate to annotate
/usr/bin/exonerate --model protein2genome -q h_annuus_proteome.fasta -t
genome.scf.fasta --showalignment no --showtargetgff --fsmmemory 500 --
seedrepeat 100 > exonerate_mem_seed.out

# convert the exonerate output to gff file
grep 'exonerate:protein2genome:local' exonerate.out | grep -v
'exonerate:protein2genome:local ' > exonerate_out.gff

####################### runExonerate.sh ########################
```

SCRIPT / PROGRAM NAME:

annotationBLAST.sh

PURPOSE:

Use tBLASTn to annotate the scaffolds from the assembled genome based on the proteome for *H. annuus* and *A. thaliana.*

APPLICATION IN THIS PROJECT:

One of the annotation methods used for this project was a translated BLAST to compare the genomic sequence scaffolds to the reference proteomes of *H. annuus* and *A. thaliana* to identify the genes in the assembly. This method was used as an alternative to exonerate because of computational challenges associated with exonerate.

USAGE INFORMATION:

This script requires three input files, the genome.scf.fasta file containing the assembled scaffolds in FASTA format, the proteome for *H. annuus* in FASTA format, and the proteome for *A. thaliana* in FASTA format. Execute this script from a directory containing all three of those files.

```
./runExonerate.sh
```

```
###################### annotationBLAST.sh ######################

#!/bin/bash

#Make blast database from contigs in scaffold
/usr/local/bin/blastplus/makeblastdb -in genome.scf.fasta -input_type fasta -
dbtype nucl -out full_assembly_blast_db -title "Full Assembly Nt Blast DB"


#Arabidopsis blast
/usr/local/bin/blastplus/tblastn -db full_assembly_blast_db -query
arabidopsis_thaliana_proteome.fasta -out arabidopsis_blast.out -num_threads 6
-outfmt "6 qseqid sseqid pident nident length slen qlen mismatch gaps evalue
bitscore sstart send score"

#Sunflower blast
/usr/local/bin/blastplus/tblastn -db full_assembly_blast_db -query
h_annuus_proteome.fasta -out sunflower_blast.out -num_threads 6 -outfmt "6
qseqid sseqid pident nident length slen qlen mismatch gaps evalue bitscore
sstart send score"

###################### annotationBLAST.sh ######################
```

SCRIPT / PROGRAM NAME:

blast2gff.R

PURPOSE:

The purpose of this script is to convert the tabular output from tBLASTn to GFF format for the tBLASTn against the *H. annuus* proteome and for the tBLASTn against the *A. thaliana* proteome.

APPLICATION IN THIS PROJECT:

The final annotation format used for this project was the gff format. First, the blast hits are filtered using the thresholds described in the methods, and then the tabular blast output needs to be converted to gff format, which contains eight specific columns about the annotation and a ninth annotation that is descriptive and can be a little more abstract. A GFF file was created by converting both outputs and appending them together.

USAGE INFORMATION:

The input required for this R script are the blast output files. Execute the commands from within RStudio.

```
########################## blast2gff.R ############################

library(data.table)

# function to convert blast datatable to gff format
makegff = function(dt, annoSp){
  gff = c()
  for(i in 1:nrow(dt)){
    seqname = dt$sseqid[i]
    source = "blast"
    feature = "gene"
    if(dt$sstart[i] > dt$send[i]){
      start = dt$send[i]
      end = dt$sstart[i]
      strand = "-"
    } else{
      start = dt$sstart[i]
      end = dt$send[i]
      strand = "+"
    }
    score = dt$pident[i]
    frame = "."
    attribute = paste0("annoSp=", annoSp, "; geneSymbol=", dt$geneName[i],
                       "; proteinName=", dt$proteinName[i], "; accessionNo=",
dt$accNo[i],
                       "; e-value=", dt$evalue[i])
    row = c(seqname, source, feature, start, end,
score,strand,frame,attribute)
    gff = rbind(gff, row)
    if(i %% 1000 == 0){
      cat ("i is ", i, "\n")
    }
  }
  data.table(gff)
}

# read in and format blast output
sunflower.out = fread("sunflower_blast.out", sep = "\t", header = F,
stringsAsFactors = F)
arabidopsis.out = fread("arabidopsis_blast.out", sep = "\t", header = F,
                        stringsAsFactors = F)
comp.out = fread("arabidopsis_blast_comp.out", sep = "\t", header = F,
                 stringsAsFactors = F)

colnames(sunflower.out) = c("qseqid", "sseqid", "pident", "nident", "length",
"slen",
                            "qlen", "mismatch", "gaps", "evalue", "bitscore",
"sstart",
                            "send", "score")
colnames(arabidopsis.out) = c("qseqid", "sseqid", "pident", "nident",
"length", "slen",
                              "qlen", "mismatch", "gaps", "evalue",
"bitscore", "sstart",
                              "send", "score")

########################## blast2gff.R ############################
```

```
########################## blast2gff.R ############################

colnames(comp.out) = c("qseqid", "sseqid", "pident", "nident", "length",
"slen",
                       "qlen", "mismatch", "gaps", "evalue", "bitscore",
"sstart",
                       "send", "score")

# parameters for plotting
par(mfrow = c(3,1),
    cex.lab = 1.5,
    cex.main = 1.5,
    cex.axis = 1.25)

# plot distributions of evalues
hist(log(sunflower.out$evalue, base = 10), breaks = 200,
     xlab = 'log10(evalue)', main = "L. maackii assembly vs. H. annuus
proteome")
abline(v = log(10^-5), col = "red", lwd = 3, lty = 2)
hist(log(arabidopsis.out$evalue, base = 10), breaks = 200,
     xlab = "log10(evalue)", main = "L. maackii assembly vs. A. thaliana
proteome")
abline(v = log(10^-5), col = "red", lwd = 3, lty = 2)
hist(log(comp.out$evalue, base = 10), breaks = 200,
     xlab = "log10(evalue)", main = "C. himalaica assembly vs. A. thaliana
proteome")
abline(v = log(10^-5), col = "red", lwd = 3, lty = 2)

# plot distributions of lengths
hist(sunflower.out$length, breaks = 200,
     xlab = "length", main = "L. maackii assembly vs. H. annuus proteome")
abline(v = 50, col = "red", lwd = 3, lty = 2)
hist(arabidopsis.out$length, breaks = 200,
     xlab = "length", main = "L. maackii assembly vs. A. thaliana proteome")
abline(v = 50, col = "red", lwd = 3, lty = 2)
hist(comp.out$length, breaks = 200,
     xlab = "length", main = "C. himalaica assembly vs. A. thaliana
proteome")
abline(v = 50, col = "red", lwd = 3, lty = 2)

# get protein info for HA proteome
fasta_ids = readLines("header_info/HA_header_ids.txt")
fasta_genenames = readLines("header_info/HA_header_genenames.txt")
fasta_proteinnames = readLines("header_info/HA_header_proteinnames.txt")
fasta_accno = readLines("header_info/HA_header_accno.txt")
fasta_header = cbind(fasta_ids, fasta_genenames, fasta_proteinnames,
fasta_accno)
colnames(fasta_header) = c("ID", "GeneName", "ProteinName", "AccNo")
head(fasta_header)
fasta_header = as.data.table(fasta_header)
head(fasta_header)



########################## blast2gff.R ############################
```

```
########################## blast2gff.R ############################

# get protein info for AT proteome
AT_header = fread("header_info/AT_header_table.txt")
colnames(AT_header) = c("ID", "ProteinName", "GeneName", "AccNo")
AT_header = AT_header[,c("ID", "GeneName", "ProteinName", "AccNo")]


# filter hits by length and evalue thresholds
sunflower_filt = sunflower.out[length >= 50 & evalue <= 0.00001]
arabidopsis_filt = arabidopsis.out[length >= 50 & evalue <= 0.00001]

# add genename, proteinname, and accessionNo columns to filtered tables
sunflower_filt = merge(sunflower_filt, fasta_header, by.x = "qseqid", by.y =
"ID")
arabidopsis_filt = merge(arabidopsis_filt, AT_header, by.x = "qseqid", by.y =
"ID")

# remove any arabidopsis hits that have overlapping gene names with sunflower
hits
# arabidopsis_filt = arabidopsis_filt[!(GeneName %in%
unique(sunflower_filt$GeneName))]

# get top hit for each sunflower protein by lowest evalue
sunflower_best = split(sunflower_filt, by = "qseqid")
sunflower_best = lapply(sunflower_best, function(x){
  x[evalue == min(evalue)]
})
sunflower_best = do.call(rbind, sunflower_best)
nrow(sunflower_best)

# get top hit for each arabidopsis protein by lowest evalue
arabidopsis_best = split(arabidopsis_filt, by = "qseqid")
arabidopsis_best = lapply(arabidopsis_best, function(x){
  x[evalue == min(evalue)]
})
arabidopsis_best = do.call(rbind, arabidopsis_best)
nrow(arabidopsis_best)

# convert BLAST output to gff format
sunflower_gff = makegff(sunflower_best, "Helianthus annuus")
write.csv(sunflower_gff, "lonicera_maackii_HA_blast.csv", row.names = F)
arabidopsis_gff = makegff(arabidopsis_best, "Arabidopsis thaliana")
write.csv(arabidopsis_gff, "lonicera_maackii_AT_blast.csv", row.names = F)
full_gff = rbind(sunflower_gff, arabidopsis_gff)
write.table(full_gff, "lonicera_maackii_blast.gff", sep = "\t",
            row.names = F)




########################## blast2gff.R ############################
```

SCRIPT / PROGRAM NAME:

exonerate2gff.R

PURPOSE:

The purpose of this is to convert the pre-GFF formatted exonerate output to a more well formatted gff format with a more descriptive attribute column.

APPLICATION IN THIS PROJECT:

Exonerate has the option to produce GFF format, but the attribute column of that format is not as informative as the attribute column defined for the blast annotation GFF file. This script formats the exonerate GFF to match more closely to the GFF attribute of the blast annotation.

USAGE INFORMATION:

The input required for this script is the exonerate output GFF format. Execute the commands from script from within RStudio.

```
######################## exonerate2gff.R ########################

library(data.table)

# exonerate to gff
exonerate = fread("exonerate_full_seed_mem_take2.txt")
colnames(exonerate) = c("seqname", "source", "feature", "start", "end",
"score","strand","frame","attribute")

# get protein info for HA proteome
fasta_ids = readLines("header_info/HA_header_ids.txt")
fasta_genenames = readLines("header_info/HA_header_genenames.txt")
fasta_proteinnames = readLines("header_info/HA_header_proteinnames.txt")
fasta_accno = readLines("header_info/HA_header_accno.txt")
fasta_header = cbind(fasta_ids, fasta_genenames, fasta_proteinnames,
fasta_accno)
colnames(fasta_header) = c("ID", "GeneName", "ProteinName", "AccNo")
head(fasta_header)
fasta_header = as.data.table(fasta_header)
head(fasta_header)

# add more detail to the attribute column
annoSp = "Helianthus annuus"
attributes = c()
for(i in 1:nrow(exonerate)){
  if(exonerate$feature[i] == "gene"){
    sunseq = strsplit(strsplit(exonerate$attribute[i], ";")[[1]][2], "
")[[1]][3]
    geneName = fasta_header[ID == sunseq]$GeneName
    proteinName = fasta_header[ID == sunseq]$ProteinName
    accNo = fasta_header[ID == sunseq]$AccNo
    attribute = paste0("annoSp=", annoSp, "; geneSymbol=", geneName,
                       "; proteinName=", proteinName, "; accessionNo=",
accNo)
  }
  attributes = c(attributes, attribute)
}

gff = cbind(exonerate[,1:(ncol(exonerate)-1)],attributes)
colnames(gff) = c("seqname", "source", "feature", "start", "end",
"score","strand","frame","attribute")
write.table(gff, "lonicera_maackii_exonerate.gff", row.names = F, sep = "\t")

######################## exonerate2gff.R ########################
```

SCRIPT / PROGRAM NAME:

compare_annotations.R

PURPOSE:

The purpose of this script is to analyze the annotations found in the GFF file for the blast annotation method and the exonerate annotation method.

APPLICATION IN THIS PROJECT:

Two annotation methods were used for this project, which led to two different GFF files being created. The output of the two annotation methods was compared to understand the similarities and differences of the two methods. The second part of this script looks for genes of interest within both of the gff files to try to draw meaningful conclusions from the annotations.

USAGE INFORMATION:

The input for this script are the two GFF files, one for the BLAST annotation method and one for the exonerate annotation method. Execute the commands within RStudio.

```
#################### compare_annotations.R ####################

library(data.table)
library(dplyr)

# function to see if a gene is in the gff file:
get_hit_rows = function(dt, pattern){
  dt[grep(pattern, dt$attribute)]
}

# function to get the unique genes
unique_genes = function(dt){
  genes = apply(dt,1,function(x){
    strsplit(x[9], ";")[[1]][4]
  })
  unique(genes)
}

# read in the gff files
blast = fread("lonicera_maackii_blast.gff")
colnames(blast) = c("seqname", "source", "feature", "start", "end",
"score","strand","frame","attribute")
exonerate = fread("lonicera_maackii_exonerate.gff")
colnames(exonerate) = c("seqname", "source", "feature", "start", "end",
"score","strand","frame","attribute")

# split blast annotations by AT and HA
blast_AT = blast[grep("annoSp=Arabidopsis thaliana", blast$attribute)]
blast_HA = blast[grep("annoSp=Helianthus annuus", blast$attribute)]

# compare the number of genes
# unique AT genes found in blast
length(unique_genes(blast_AT))
# unique HA genes found in blast
length(unique_genes(blast_HA))
# unique HA genes found in exonerate
length(unique_genes(exonerate))
# HA genes found in blast but not in exonerate
HA_genes = unique_genes(blast_HA)
exonerate_genes = unique_genes(exonerate)
length(HA_genes[!(HA_genes %in% exonerate_genes)])
# HA genes found in exonerate but not in blast
length(exonerate_genes[!(exonerate_genes %in% HA_genes)])
exonly = exonerate_genes[!(exonerate_genes %in% HA_genes)]
for(i in 1:length(exonly)){
  cat(exonerate[grep(exonly[i], exonerate$attribute)]$attribute[1], "\n")
}

# compare alignment length
# all alignments
blast_length = abs(blast_HA$start - blast_HA$end)
mean(blast_length)
median(blast_length)


#################### compare_annotations.R ####################
```

```
##################### compare_annotations.R ####################

exonerate_genes_only = exonerate[feature == "gene"]
exonerate_length = abs(exonerate_genes_only$start - exonerate_genes_only$end)
mean(exonerate_length)
median(exonerate_length)
par(mfrow = c(1,1))
boxplot(log(blast_length), log(exonerate_length), names = c("tBLASTn",
"Exonerate"),
        ylab = "ln(Alignment Length)", xlab = "Annotation Method")
wilcox.test(exonerate_length, blast_length, alt = "g")


# accession numbers for the genes of interest:
interesting.gene.ids = as.list(c('A0A251UTT7', # GGPPS - putative
geranylgeranyl pyrophosphate synthase
                                 'A0A251UA19', # CRTSO - prolycopene
isomerase
                                 'Q8H0Q6', # zds - zeta-carotene desaturase
                                 'A0A251TZI8', # LCYB - putative lycopene
beta cyclase
                                 'A0A0K3A5X2', # CCD4-L - carotenoid cleavage
dixoygenase 4-like protein
                                 'A0A1Y3BUK2', # CCD8A - putative carotenoid
cleavage dioxygenase 8
                                 'A0A251SWS7', # CCD7 - putative carotenoid
cleavage dioxygenase 7 protein
                                 'A0A251VH23', # CCD8B - putative carotenoid
cleavage dioxygenase
                                 'A0A251SKA0', # PALY - phenylalanine
ammonia-lyase
                                 'A0A251SRU1', # phenylalanine ammonia lyase
                                 'A0A251SRY0', # phenylalanine ammonia lyase
                                 'A0A251SUN9', # Phenylalanine ammonia lyase
                                 'A0A251TUG3', # PAL1 - phenylalanine ammonia
lyase
                                 'A0A251UCP2', # PAL1 - phenylalanine ammonia
lyase
                                 'A0A251VH89', # PALY - phenylalanine ammonia
lyase
                                 'A0A251VJ15', # phenylalanine ammonia lyase
                                 'A0A251TB89', # LOX15 - lipoxygenase
                                 'A0A251TKU3', # LOX5 - lipoxygenase
                                 'A0A251U0R8', # LOX31 - lipoxygenase
                                 'A0A251V5M7', # LOXA - lipoxygenase
                                 'A0A251VIG0', # LOXC1 - lipoxygenase
                                 'A0A251VIG5', # LOX2 - lipoxygenase
                                 'A0A251VPY6', # LOX21 - putative linoleate
13s-lipoxygenase 2-1 protein
                                 'A0A251RSX1', # JAR1, putative auxin
responsive GH3 family protein
                                 'AT4G14210', # Q07356 - 15-cis-phytoene
desaturase (PDS)
                                 'AT5G52570', # beta-carotene 3-hydrolase 2,
chloroplastic
##################### compare_annotations.R ####################
```

```
                                'Q93ZN9', # LL-diaminopimelate
aminotransferase, chloroplastic
                                'P05466')) # EPSPS
get_hit_rows = function(dt, pattern){
  dt[attribute %in% grep(pattern, dt$attribute, value = T)]
}


blast.genes.found = lapply(interesting.gene.ids, function(x){
  out = get_hit_rows(blast, x)
  out[,c("seqname", "attribute")]
})
blast.genes.found = do.call(rbind, blast.genes.found)

exonerate.genes.found = lapply(interesting.gene.ids, function(x){
  out = get_hit_rows(exonerate_genes_only, x)
  out[,c("seqname", "attribute")]
})
exonerate.genes.found = do.call(rbind, exonerate.genes.found)
```